

Utveckling av testsystem för säkerhetsautomationssystem i kraftvärmeverk

Marcus Ahlvik

Examensarbete för ingenjörsexamen (YH)

Utbildningsprogrammet för elektroteknik

Vasa 2012



EXAMENSARBETE

Författare: Marcus Ahlvik
Utbildningsprogram och ort: Elektroteknik, Vasa
Inriktningsalternativ: Automationsteknik
Handledare: Matts Nickull

Titel: *Utveckling av testsystem för säkerhetsautomationssystem i kraftvärmeverk*

Datum 12.4.2012

Sidantal 44

Sammanfattning

Detta examensarbete har gjorts på uppdrag av Vaasa Engineering Oy. Examensarbetet omfattar utveckling av programvaran till ett testsystem för fabrikstest av säkerhetsautomationsenheter. Syftet med arbetet är att uppdragsgivaren ska få ett snyggt, effektivt och användarvänligt sätt att testa säkerhetsautomation till deras värmekraftverksprojekt. Utvecklingen av testsystemet är uppdelat i två separata examensarbeten. Detta examensarbete omfattar planering och programmering av programvaran.

Målet med testsystemet var att få ett system där man effektivt kan testa säkerhetsautomatiken genom att simulera förändringar i någon ingenjörsenhet. Man ska även kunna simulera analoga och digitala signaler samt läsa in digitala signaler. Till testsystemet behövdes även ett effektivt verktyg att importera data med som max- och minvärde, tag-namn samt beskrivningar. Till testsystemet behövdes även en användarmanual för att göra det enklare för nya användare att börja använda testsystemet. Testsystemet förverkligades med en PLC från Siemens. Programmeringen gjordes med Siemens nya verktyg TIA Portal. Detta projekt är första i vilket TIA Portalen används av uppdragsgivaren och fungerar därmed som ett pilotprojekt.

Testsystemet blir det primära verktyget för test av säkerhetsautomation vid uppdragsgivarens systemtester. Importverktyget, som skapades i Microsoft Excel, möjliggör effektivt arbete före själva testet. Resultatet av examensarbetet blev ett användarvänligt och effektivt testsystem.

Språk: svenska Nyckelord: FAT, TIA Portal, kraftvärmeverk, testsystem

Tillgängligt: Theseus.fi och i Tritonia, Vasa vetenskapliga bibliotek.

OPINNÄYTETYÖ

Tekijä: Marcus Ahlvik
Koulutusohjelma ja paikkakunta: Sähkötekniikka, Vaasa
Suuntautumisvaihtoehto: Automaatiotekniikka
Ohjaaja: Matts Nickull

Nimike: *Testausjärjestelmän kehitys lämpövoimalaitoksen turva-automaatiojärjestelmän testaukseen*

Päivämäärä 12.4.2012

Sivumäärä 44

Tiivistelmä

Tämä opinnäytetyö on tilattu Vaasa Engineering Oy:ltä. Opinnäytetyö käsittelee testausjärjestelmän ohjelmiston kehitystä ja ohjelmointia. Työn tarkoitus on että työnantaja saa hyvännäköisen, tehokkaan ja käyttäjäystävällinen testausjärjestelmän. Testausjärjestelmän kehitys on jaettu kahteen osaan, laitteistosuunnitteluun ja ohjelmistosuunnitteluun. Tämä työ käsittelee ohjelmiston kehitystä.

Työn tavoite on kehittää testausjärjestelmän jolla voi tehokkaasti simuloida muutoksia jossakin insinööriyksikössä. Testausjärjestelmällä pitää myös olla mahdollisuus simuloida analogisia ja digitaalisia lähtöjä sekä näyttää digitaalisten tulosten tilat. Testausjärjestelmä tarvitsee myös tehokkaan tuontityökalun, joka tuo tietoa Excel-tilauksesta, esim. Tag-nimi, max- ja min-arvoja. Testausjärjestelmälle tarvitaan myös käsikirja, joka tekee aloituksen uusille käyttäjille helpommaksi. Testausjärjestelmä on toteutettu Siemensin PLC:llä. PLC ja käyttöliittymä ohjelmoitiin Siemensin uudella työkalulla, TIA Portal. Tämä projekti on työnantajan ensimmäinen työ, jossa käytetään TIA Portaalia.

Tuontityökalu, joka on tehty Microsoft Excelissä, mahdollistaa työn tehokkuuden ennen varsinaista testausta. Opinnäytetyön tulos on käyttäjäystävällinen ja tehokas testausjärjestelmä.

Kieli: ruotsi Avainsanat: FAT, TIA Portal, lämpövoimalaitos, testausjärjestelmä

Arkistoidaan: Theseus.fi ja Tritoniassa, Vaasan tiedekirjasto.

BACHELOR'S THESIS

Author: Marcus Ahlvik
Degree Programme: Electrical Engineering
Specialization: Automation technology
Supervisor: Matts Nickull

Title: *Development of a test system for testing safety automation in thermal power plants*

Date 12.4.2012

Number of pages 44

Summary

This thesis has been carried out on behalf of Vaasa Engineering Oy. The thesis covers the development of software for a test system for factory acceptance testing of the safety automation systems. The purpose of this work is to provide Vaasa Engineering with an efficient and user friendly way to test the safety automation of their thermal power plant projects. The development of the test system is divided into two separate theses. This thesis covers the planning and programming of the software.

The goal of the test system was to have a system where you can effectively test the safety automation by simulating changes in any engineering unit. The test system should also be able to simulate analog and digital signals and retrieve digital signals. The test system also needed an effective tool to import data like maximum and minimum values, tag names and descriptions. A user manual was also needed to make it easier for new users to start using the test system. The test system was made with a Siemens PLC. The programming was done with Siemens' new tool, TIA Portal. This project is the first in which TIA Portal is used by Vaasa Engineering, thereby functioning as a pilot project.

The test system is going to be the primary tool for testing the safety automation during the client's Factory Acceptance Tests (FAT). The import tool, created in Microsoft Excel, enables efficient work before the actual FAT. The result of the thesis was a user friendly and efficient test system.

Language: swedish Key words: FAT, TIA Portal, thermal power plants, test system

Filed at: Theseus.fi and in Tritonia Academic Library, Vaasa.

Innehållsförteckning

1	Inledning	1
2	Syfte	2
3	Kraftvärmeverk	3
3.1	Kraftverkets huvuddelar	4
3.2	Kraftverksautomation	7
3.3	Säkerhetsrelaterade system (SRS)	8
4	Systemtest vid leverantören (FAT)	11
4.1	Förberedelser	11
4.2	Testets genomförande	13
4.3	Slutförande	16
5	PLC	17
5.1	PLC-arkitektur	17
5.2	Programmeringsspråk	19
6	HMI	20
6.1	Färganvändning	20
6.2	Konsekvent design	21
7	TIA Portal	22
8	Testsystemets förverkligande	24
8.1	Planering och programmering	24
8.1.1	Maskinvarukonfiguration	25
8.1.2	PLC	26
8.1.3	WinCC	35
8.2	Importverktyg	39
9	Resultat och tolkning	41
10	Kritisk granskning och diskussion	43
	Källförteckning	44

Figurförteckning

<i>Figur 1. Kondenskraftverkens cirkulationskretsar och huvuddelar.</i>	<i>4</i>
<i>Figur 2. Blockschema över programuppbyggnaden.</i>	<i>25</i>
<i>Figur 3. Översikt på maskinvarukonfigurationen.</i>	<i>26</i>
<i>Figur 4. Programblock för analog utgång.</i>	<i>28</i>
<i>Figur 5. Jämförelse ifall värde ut är större än max-värde.</i>	<i>29</i>
<i>Figur 6. Beräkning av mätområde.</i>	<i>29</i>
<i>Figur 7. Värdet på utgången sätts till minimivärdet.</i>	<i>30</i>
<i>Figur 8. Sätter värdet ut till 25 % av mätområdet när villkoren är uppfyllda.</i>	<i>30</i>
<i>Figur 9. Stannar rampfunktionen ifall den är igång när kanalen sätts i auto-läge.</i>	<i>31</i>
<i>Figur 10. Värdet på rampfunktionen sätts till ingångsvärdet när rampen stängs av.</i>	<i>31</i>
<i>Figur 11. Rampblocket lånat av Vaasa Engineering Oy.</i>	<i>32</i>
<i>Figur 12. Värdet på rampen konverteras från real till integer.</i>	<i>32</i>
<i>Figur 13. Programblock för digital utgång.</i>	<i>33</i>
<i>Figur 14. Sätter värdet från auto-biten till status-biten.</i>	<i>33</i>
<i>Figur 15. Sätter värdet från auto-biten till status-biten.</i>	<i>34</i>
<i>Figur 16. Sätter rätt värde på status-biten.</i>	<i>34</i>
<i>Figur 17. Sätter rätt värde på status-biten.</i>	<i>34</i>
<i>Figur 18. Status från digital ingång förs över till morsvarande DB-tag.</i>	<i>35</i>
<i>Figur 19. Layout över analoga utgångar.</i>	<i>36</i>
<i>Figur 20. Användargränssnitt för analog utgång.</i>	<i>37</i>
<i>Figur 21. Layout över digitala utgångar.</i>	<i>38</i>
<i>Figur 22. Layout över digitala ingångar.</i>	<i>38</i>
<i>Figur 23. Användargränssnitt för digital utgång.</i>	<i>39</i>
<i>Figur 24. Användargränssnitt för digital ingång.</i>	<i>39</i>
<i>Figur 25. Importverktyg analog utgång.</i>	<i>40</i>
<i>Figur 26. Uppdragsgivarens tidigare testsystem.</i>	<i>41</i>
<i>Figur 27 Testsystemet med dess snabbkontakter.</i>	<i>42</i>
<i>Figur 28 Maskinvaran i testsystemet.</i>	<i>42</i>

Ordförklaring

Byte	Digital enhet bestående av 8 bitar.
CPU	Central Processing Unit
DCS	Distributed Control System, ett styrsystem som inte är centraliserad till ett system
FAT	Factory Acceptance Test
FB	Funktionsblock
FC	Funktion
Longword	Digital enhet bestående av 32 bitar.
Kavitation	Ett fenomen då luftbubblor uppstår till följd av att trycket minskar i en vätska. (vätskan kokar)
OB	Organisationsblock
PLC	Programmable Logic Controller
RAM	Random Access Memory
Redundant	Ett redundant system betyder att det finns en dublicering av någon kritisk component, t.ex två servrar med exakt samma information där backup server tar över ifall primära får något fel.
ROM	Read Only Memory
SRS	Safety Related System, säkerhetsrelaterade system.
STEP 7	Programmeringsverktyg för programmering av PLC.
TIA Portal	Totally Integrated Automation Portal, Siemens nyaste verktyg för programmering av PLC och HMI.
WinCC	Verktyg för utformning av användargränssnitt.
Word	Digital enhet bestående av 16 bitar.

Förord

Jag vill tacka Vaasa Engineering Oy som har gett mig möjligheten att utföra ett intressant och lärorikt examensarbete. Speciellt vill jag tacka Johan Lång från Vaasa Engineering som fungerat som min mentor under hela projektet.

Dessutom vill jag även tacka min sambo för att hon gett mig möjligheten att utföra examensarbetet genom att ta hand om vår son under de stunder jag behövt för att slutföra projektet och alla andra som har medverkat i projektet.

Marcus Ahlvik

1 Inledning

Detta examensarbete är ett uppdrag av Vaasa Engineering. Uppdraget består av att planera och programmera ett logikbaserat testsystem för test av säkerhetsautomation till kraftvärmeverk. Uppdraget utförs tillsammans med en kollega och går ut på att utveckla ett logikbaserat testsystem som ska underlätta och förbättra test av säkerhetsautomation. Arbetet har delats upp i två delar. Den ena delen av arbetet omfattar planering av maskinvaran, vilket en kollega utför, medan detta examensarbete omfattar planering och programmering av programvaran.

Syftet med detta examensarbete är att kartlägga hur testsystem för ett säkerhetsautomationssystem i kraftvärmeverk kan utvecklas. Detta görs för att underlätta och effektivisera test av säkerhetsautomation vid uppdragsgivarens fabrikstester. I resultatet av arbetet presenteras utvecklingen av ett logikbaserat testsystem för test av säkerhetsautomation i kraftvärmeverk.

Alla kraftverksprojekt är olika, därför måste programvaran vara lätt att modifiera så den lämpar sig för samtliga projekt. Testsystemet ska placeras i ett relativt litet skåp på hjul så det lätt kan tas fram och ställas undan. Programvaran kommer att utvecklas med Siemens nya programmeringsverktyg, TIA Portal. Detta projekt är det första vid Vaasa Engineering där verktyget används.

Vaasa Engineering Oy levererar automations- och elektrifieringslösningar för distribuering, produktion, överföring och användning av elektricitet främst inom Europa. Till VEO-koncernen hör Vaasa Engineering, Vaasa Kojestot och Vaasa Service. Vaasa Kojestot tillverkar elcentraler, medan Vaasa Service som sköter installation och underhåll. Vid avdelningen där examensarbetet utförts planeras kraftvärmeverkens elektrifiering och automation. Till de planerade kraftverken planeras även säkerhetsautomationssystem med logikbaserad automation som säkerställer kraftverkets säkerhet. Detta system behöver testas innan det levereras till kundens kraftverk. Testningen görs med FAT (Factory Acceptance Test). FAT har tidigare utförts med relativt primitiva testmetoder som har bestått av testapparat med fastmonterade brytare och potentiometrar. Dessa brytare och potentiometrar har simulerat digitala och analoga in- och utgångar till logiken.

2 Syfte

Syftet med detta examensarbete är att kartlägga hur testsystem för ett säkerhetsautomationssystem i kraftvärmeverk kan utvecklas. Detta görs för att underlätta och effektivisera test av säkerhetsautomation vid uppdragsgivarens fabrikstester. I resultatet av arbetet presenteras utvecklingen av ett logikbaserat testsystem för test av säkerhetsautomation i kraftvärmeverk.

3 Kraftvärmeverk

Idag används flera olika slag av värmekraftverk och kraftvärmeverk. Skillnaden mellan värmekraftverk och kraftvärmeverk är att i värmekraftverk producerar man endast elektrisk energi, medan man i ett kraftvärmeverk även tar tillvara spillvärmen. Värmen som tas till vara används oftast till fjärrvärme som leds ut i nätet till kunderna. Verkningsgraden hos ett värmekraftverk ligger runt 50 %, medan det hos ett kraftvärmeverk kan ligga runt 90 %. Tidigare var kol det vanligaste bränslet men någon gång runt 1950 började oljan ta över. Oljan var det vanligaste bränslet framtill 1980. Oljekriserna på 70-talet gjorde kraftiga prishöjningar på råoljan och därmed var det inte lika lönsamt längre att använda olja. Idag är det vanligt att man använder sig av något biobränsle, eftersom vi är mer medvetna om växthuseffekterna. En vanlig typ av värmekraftverk är kondenskraftverk, av kraftvärmeverk är mottryckskraftverken vanligast.

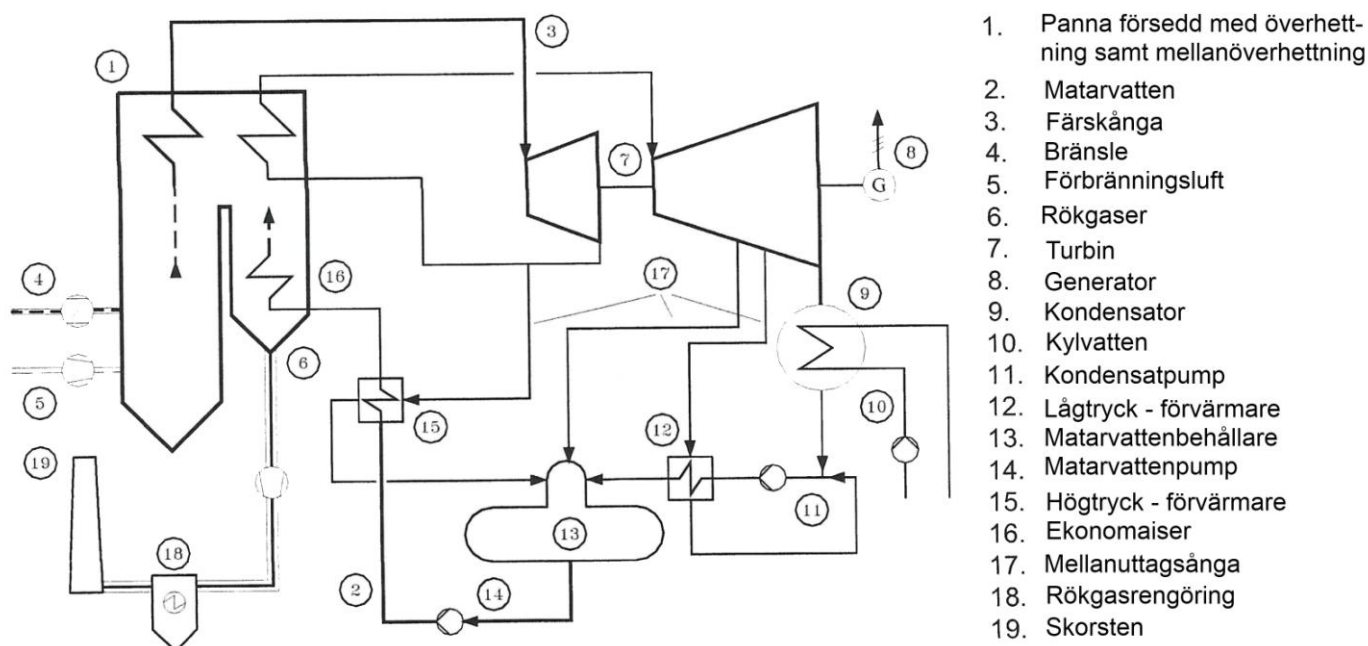
Ur automationssynvinkel är kraftverk en ganska snabb process, där effekten måste kunna regleras enligt energibehovet på ett relativt snabbt sätt. Kraftverkens upp- och nedkörningar måste även ske tillräckligt snabbt. I kraftverk finns det ett flertal direkta eller indirekta regleringar som ger upphov till komplexa helheter. Kraftverken som använder sig av fasta bränslen kan orsaka problem vid regleringen. Förutom det ovannämnda är t.ex. ångturbinens inkommande ångas tryck och temperaturer reglertoleranser ganska små. Detta är p.g.a. av att man försöker nå en så hög verkningsgrad som möjligt.

I mottryckskraftverk tas både värme- och elenergi till vara, där värme är primärprodukten medan elektricitet är sekundär. I ett kondenskraftverk tas kylvattnet främst från närliggande sjö eller hav och därmed tas inte värmen till godo. Mottryckskraftverk har i stora drag samma huvudkomponenter som i figur 1 bara att värmeåtervinningskomponenter såsom värmeväxlare inte finns. Eftersom skillnaderna i huvudkomponenterna är ganska små tas nedan främst kondenskraftverkets delar upp. (Joronen, Kovács, & Majanne, 2007).

3.1 Kraftverkets huvuddelar

För att illustrera komplexiteten i ett värmekraftverk presenteras nedan de olika delarna i ett kondenskraftverk noggrannare.

Ur figur 1 framgår kondenskraftverkens cirkulationskretsar och huvuddelar.



Figur 1. Kondenskraftverkens cirkulationskretsar och huvuddelar.

Kylvattensystem

Kraftverk är i behov av kylvatten till flera delar i anläggningen. De delar som är i största behov av kylvatten är:

- turbinkondensator
- hjälpkondensator
- kylning av avfallsvattenbehållare
- övriga delar i turbinen som kräver kylning.

Anläggningar som är belägna vid kusten använder sig av havsvatten i turbinkondensatorerna. Om man använder havsvatten i turbinkondensatorn måste man välja ett material som tål havsvatten. Saltvattnet gör att metaller som kommer i kontakt med det lättare börjar oxidera och därmed försämras dess hållfasthet. Titan är en ädlare

metall än järn och därför oxiderar den inte lika lätt. Titan är en metall som kan användas i en turbinkondensator där det används havsvatten. Vid andra delar där kylvatten behövs används helst sötvatten för att undvika oxidering. (Joronen, Kovács, & Majanne, 2007).

Matarvattensystem

Matarvattensystem omfattar matarvattenbehållaren, matarvattenpumpen samt tillhörande rörsystem. Matarvattenbehållarens uppgift är bl.a. att fungera som lagstadgade matarvattenlager, avlägsna gas från tilläggsvattnet och kondensatet samt fungera som belastningsutjämnare. Matarvattenpumpens uppgift är att höja matarvattnets tryck till panntrycket. För att pumpen ska fungera effektivt bör man undvika kavitation. Meningen med förvärmning av matarvattnet är att höja kraftverkets verkningsgrad. Lågtrycksförvärmare är placerade mellan kondensatorn och matarvattenbehållaren, och med dessa förvärms kondensat från turbinen samt processen. Högtrycksförvärmarna är placerade efter matarvattenpumparna. Matarvattnets sista förvärmningsfas är rökgasförvärmning. Matarvattnets rökgasförvärmare, s.k. ekonomaiser, är placerad efter rökgaskanalens överhettare. (Joronen, Kovács, & Majanne, 2007).

Kondensatsystem

I kondensatkraftverk bildas största delen av kondensatet i turbinens kondensator, varifrån det leds till kondensatbehållaren. I fjärrvärmeverk sker kondensering vid fjärrvärmewäxlare, där värme överförs från lågtrycksånga till fjärrvärmevattnet. Det uppsamlade kondensatet pumpas till matarvattenbehållaren. I returlinjen värms kondensatet i lågtrycksförvärmaren. (Joronen, Kovács, & Majanne, 2007).

Panna

Pannan är kraftverkets mest centrala del både med tanke på verkningsgrad och utsläpp. Den har även i användnings- och underhållssyfte stor betydelse. Pannans delsystem och dess uppgift:

Bränslematning:	Bränslemängden styrs enligt anläggningens effektbehov. Vid kondensatkraftverk bestäms effektbehovet enligt behovet av eleffekt.
-----------------	---

Bränslets förbränning:	Uppehållande av förbränningens verkningsgrad
------------------------	--

- Inmatning av bränslet enligt energibehovet.

- Jämn förbränning är speciellt viktigt vid förbränning av fasta bränslen.
- Rätt luft-bränsle förhållande för optimal förbränning.

Vattnets förångning: Uppehållande av tryck och temperatur

- För hög temperatur förkortar rörens livslängd.
- För lågt tryck eller för stora tryckförluster minskar anläggningens effekt och därmed försämras verkningsgraden.

Sotning: Värmeöverföringsytan sotas regelbundet för att upprätthålla en god verkningsgrad på pannan.

Förbränningsluftsystem

Förbränningsluftsystemet omfattar kanaler, fläktar och förvärmare. För att kunna kontrollera förbränningsprocessen tillförs pannan luft i flera faser som primär- och sekundärluft. Primär- och sekundärluftfläktarna genererar det behövliga trycket i kanalerna för att skapa en luftström. Förbränningsluften till processen tas från pannrummets överdel, därigenom tas en del av strålningsvärmeförluster till godo. Den egentliga förvärmning av förbränningsluften görs med rökgasförvärmare, som är den sista värmeöverföringen i rökgaskanalen. Ofta finns det även i pannor en ångförvärmare för uppvärmning av förbränningsluften som används vid låga effektbehov för att undvika för kraftig nedkylning av rökgaserna. Ifall rökgasens temperatur faller under svavelsyrans kondenspunkt kondenseras svavelsyran och orsakar korrosion i kanalerna. (Joronen, Kovács, & Majanne, 2007).

Ångturbin

Den värmeenergi som ångan innehåller omvandlas till mekanisk energi genom att expandera ångan genom turbinen. Expanderingsprocessen av ångan sker ofta i flera zoner efter varandra. En vanlig turbintyp är impulsturbinen. Impulszonen omfattas av stillastående munstyck och roterande vingar. Ångan expanderar genom munstycket, och då ökar dess hastighet. En ökning av hastigheten orsakas av ett minskat tryck och en ökad volym. Efter munstycket träffar ångan roterande vingar där ångans rörelseenergi överförs till roterande vingar. När ångan träffar roterande vingar minskar ångans hastighet. I impulsturbinen minskar

ångtrycket i munstycket men hålls konstant genom rotern. (Joronen, Kovács, & Majanne, 2007).

3.2 Kraftverksautomation

Automationssystem har alltid haft en central roll i värmekraftverk, men kraftverk ställer speciella krav på automationssystemet. Dessa uppkommer från energiproduktionens säkerhetskrav eftersom systemets förmåga att kontrollera störnings- och farosituationer både kräver hög kapacitet och hög automationsgrad. I kraftverk används i huvudsak samma automationssystem som i processindustrin, men i förverkligande av systemen framhävs säkerhetskraven. Dessa krav gallrar naturligt ut vissa system på marknaden. Ett exempelkrav som är speciellt vanligt i större kondenskraftverks är det höga kravet på svarstid. I kraftverksautomation är det mycket vanligt med konservativt tänkande gällande ny teknik, man använder sig helst av äldre beprövad teknik och lösningar och vill ogärna vara först med att testa nya systemlösningar. Som exempel har fältbuslösningar vuxit fram mycket långsammare än i den övriga processindustrin. Dessutom har denna teknik begränsats nästan enbart till nyare industrikraftverk, där frekvensomvandlare och/eller motorstyrningar kopplas ihop med systemet med t.ex. Profibus DP-bus. (Joronen, Kovács, & Majanne, 2007).

Kraftverksautomation förverkligas sällan i ett enda system. Vanligtvis skaffas ett s.k. huvudautomationssystem, där största delen av övervakning, styrning och reglering förverkligas. Huvudautomationssystem är vanligtvis av DCS-typen. Till huvudautomationssystem ansluts givetvis alla mätningar, styrningar och regleringar som hör till huvudprocessen. Med huvudprocessen menas i kraftverksprocessen allt från bränsleförvaringssilon eller -behållare till kondensat-, kylvattens- och fjärrvärmeapparatur. Kraftverksprocessen innefattar även delprocesser och anläggningar som har egna styrsystem. Dessa är exempelvis gas- och ångturbiner, dess generatorer samt rökgasens reningsanläggning, såsom el- och slangfilter. De här ansluts vanligtvis till huvudautomationssystemet, på så sätt att man kan övervaka och styra dem från kontrollrummet. (Joronen, Kovács, & Majanne, 2007).

Förutom dessa tidigare nämnda finns oftast även en del sid-, och hjälpprocesser, vars automation kan förverkligas endera i huvudautomationssystemet, som enskilt programmerad logik eller med något annat system. Dessa processer är exempelvis:

- vattenbehandlingsanläggningen (tilläggsvattens framställning och kondensat behandling)
- mottagning och transportapparat för fast bränsle
- pneumatisk transportapparat (t.ex. flygasktransportörer)
- VVS-apparat.

För de tidigare nämnda delprocesser lär det av ekonomiska skäl tala för enskilda system, medan det för helhetskontroll samt underhåll av automationen igen talar för förverkligande i huvudautomationssystemet. Ifall enskilda system skaffas ansluts de i alla fall till huvudsystemet så att man från kontrollrummet kan övervaka och styra processerna. (Joronen, Kovács, & Majanne, 2007).

En del av kraftverksautomationen är säkerhetsrelaterade system (SRS), som bildas från huvudautomations- samt andra system. I en del automationssystem bildas säkerhetssystem av säkerhetscertifierade komponenter, som kan placeras i automationsskåp eller tillsammans med annan elektronik. (Joronen, Kovács, & Majanne, 2007).

Med hjälp av processinformationssystem kan man i kraftverk utföra insamling av processhistorik och mätdata samt behövliga balans- och driftsekonombereäkningar. Med dessa kan man sedan skapa olika rapporter för internt bruk eller t.ex. skapa en utsläppsrapport till myndigheterna. Till processinformationssystemet kan man även spara alarm- och händelseinformation för längre tider, varvid störningstillfällen kan effektivare analyseras i efterhand. (Joronen, Kovács, & Majanne, 2007).

3.3 Säkerhetsrelaterade system (SRS)

Säkerhetsrelaterade system är ett skyddssystem som avbryter processen eller styr den till ett säkert läge ifall processen hamnar i farosituation. SRS omfattar hela skyddskedjan såsom givare, logik och styrdon. SRS fungerar enskilt från den egentliga driftsautomationen. SRS styr inte aktivt processen utan övervakar den och utför vid behov åtgärder. Av säkerhetsrelaterade system krävs att:

- Systemet vid en allvarlig störningssituation kör ner pannan eller styr den till säkert tillstånd, men den får inte heller orsaka onödiga nedkörningar som kan orsaka säkerhetsrisker eller för pannan skadliga nedkörningar.

- Systemet med hög sannolikhet fungerar felfritt i sådana farosituationer som eventuellt endast inträffar en gång under anläggningens livscykel.

I kraftverk förverkligas med säkerhetsautomation typiskt följande skydd:

- pannskydd
- brännarskydd
- ångturbinens övervarvskydd och ångturbinens övriga skydd
- högtryckssidans övertrycksskydd i pannor med mellanöverhettning
- gasturbinskydd.

SRS testas före ibruktagning så noggrant som möjligt. För att säkerhetsnivån ska kunna upprätthållas under hela livscykeln måste man även periodiskt testa dess funktion. Typisk testintervall är ett år. (Joronen, Kovács, & Majanne, 2007).

SIL (Safety Integrity Level)

Ett allmänt accepterat sätt vid förverkligande av SRS är att följa den internationella standarden EN-61508. Standarden utgör en modell för funktionell säkerhetslivscykel. Modellen innehåller allt från tidigt skede i planeringen ända till processens tas ur bruk, hur man kan uppnå och upprätthålla en tillräcklig nivå på säkerheten under hela livscykeln. Ett mål att sträva efter är att man för samtliga säkerhetsfunktioner ska definierar säkerhetens integritetsnivå. Med säkerhetens integritetsnivå menas hur säkert en säkerhetsfunktion utför den uppgift som den har. Desto högre integritetsnivå desto säkrare är det att systemet utför dess uppgift. (Joronen, Kovács, & Majanne, 2007).

Säkerhetsfunktioner i kraftverk som förverkligas i SRS

Pannskydd omfattar pannans bränsletillförsel vid olika farosituationer. Typiska sådana följer:

- Nödstoppsfunktion. Att kraftverket körs i ett säkert läge när någon nödstopp aktiveras sköts av SRS.
- Övervakning av rökkanalens stängning. Omfattar vanligtvis övervakning av plåtarnas, rökgasfläktarnas samt rökgasens reningsanordning.
- Övervakning av tryckökningar i eldstaden.

- Övervakning av tillräcklig förbränningsluft. I brännarskydd förverkligas det oftast genom att övervaka bränsle-luft förhållande. Det är även möjligt att stänga brännaren genom mätning av syrehalt i rökgaskanalen.
- Övervakning av förbränning. Vid förbrännarstyrningar förverkligas det genom att ha brännarspecifika flamövervakare. Övervakningen övervakar främst ifall flamman slocknar samt att flamman har uppstått en tid efter att bränsletillförseln påbörjats. Vid förbränning i statisk och cirkulerande fluidiserad bädd övervakas bäddens strömningar, tryck samt temperatur.
- Övervakning av ventilation och tändningsförsök. Vid ventilering av pannan tillförs ren luft för att få bort eventuella explosiva gaser innan tändningsförsök. Ventilering måste utföras efter ett visst antal tändningsförsök.

Utöver dessa kan man beroende på pannans typ även utföra andra funktioner såsom t.ex. eldstadens eller rökgaskanalens temperaturövervakning, tryckmätning på rökgasfäktarnas undertryckssida eller något annat pannspecifikt skydd som lyfts fram vid faro- eller riskanalyser. (Joronen, Kovács, & Majanne, 2007).

Ångturbinens övervarvsskydd är alltid med redan vid leveransen av ångturbinen. Det krävs en väldigt högt funktionssäkerhet av övervarvsskyddet. I dagens läge är det en elektronisk apparat, som i större turbiner ofta innefattar sex varvtalsgivare och deras förstärkare. Ångturbinskydd omfattar även mottrycksskydd, smörjöljans tryckskydd, övervakning av axiell rörelse och övervakning av lagrens temperatur. Mängden och omfattningen av dessa tilläggsskydd beror på tillverkaren och turbintypen. Ofta skaffas turbinens skyddssystem tillsammans med turbinen. Till skyddet kan det även tillhöra externa skyddsvillkor. Utvecklingen under den senaste tiden och uppföljningen av den funktionella säkerhetsstandarden EN 61508 har gjort att turbinskyddet ofta förverkligats med liknande teknik som i pannskydd. Turbinskyddet utför utlösning genom att stänga turbinens snabbstängningsventil. Utlösning sker genom att sänka hydrauliktrycket till snabbstängningsventilen med hjälp av olika solenoidventiler. I solenoidventiler är pådragsdonet en elektrisk spole. Oftast är solenoidventiler minst två stycken och man kan testa dess funktion vid normal drift. Ångturbinens övervarvsskydd och andra turbinskydd förverkligas som skilda system. Det enda som förenar dessa är utlösning med solenoidventilerna. (Joronen, Kovács, & Majanne, 2007).

I pannor med mellanupphettning kan man inte använda sig av övertrycksventiler på högtryckssidan, eftersom man i mellanupphettare måste ha fortlöpande ångström. Om ångturbinen stoppas leds högtrycksången via bypass-kanalen till mellanupphettaren. På så sätt fungerar bypass-kanalen som övertrycksventil. Eftersom bypass-kanalen även fungerar som reglerventil i vissa lägen finns där en del säkerhetsapparat som oberoende av andra styrsignaler öppnar ventilen ifall turbinen snabbstängs eller ångtrycket ökar. Till gasturbinens skydd hör stängning av bränsletillförselns snabbstängningsventil, om t.ex. varvtalet eller brännarkammarens temperatur stiger för högt. Skyddet kommer med leveransen av gasturbinen. (Joronen, Kovács, & Majanne, 2007).

4 Systemtest vid leverantören (FAT)

Test av systemet i leverantörensutrymmen kallas för "Factory Acceptance Test" och är den mest kritiska fasen under genomförande av ett projekt. All maskinvara och programvara som är relevant för projektet ska testas för att säkerställa att systemet fungerar i enlighet med kravspecifikationen. Kunden och tillverkaren bör slå sig ner och genom ett långt och utsträckt möte fastställa klart och tydligt vad som ska göras och vilka ansvarsområden som hör till vem. Detta görs för att säkerställa att tidtabellen håller och att systemet fungerar som det ska. Testet säkerställer även att båda parterna är överens om systemets funktionalitet och apparatur.

Den internationella standarden IEC 62381 ger förutsättningar för att snabbare få bättre resultat i förhandlingsfasen över vilka parternas ansvarsområden är. Standarden definierar förfaringssätt och specifikationer för FAT, SAT (Site Acceptance Test) och SIT (Site Integration Test). (International Electrotechnical Commission, 2006).

4.1 Förberedelser

Det kräver aktivitet från både tillverkaren och kunden vid förberedelser inför systemtest. Ett väl förberett FAT gör att själva testfasen löper smidigare.

Innan man påbörjar FAT, ska tillverkaren ha utfört ett komplett förtest. Förtest är identiskt med FAT bara med den skillnaden att den utförs av endast tillverkaren och utförs innan själva FAT. Förtestet utförs för att testa systemets funktionalitet och säkerställer att själva FAT löper flytande. (International Electrotechnical Commission 2006).

Dokumentation är nödvändig i varje skede av projektets genomförande, man bör dokumentera från varje fas och aktivitet. Olika parter behöver olika slags information eller synpunkter av samma information, beroende på vad som lämpar sig bäst för deras syfte. IEC 61355 är en standard för klassificering och design av dokument för anläggningar, system och utrustning. Ett mål med standarden är att etablera en metod för bättre kommunikation mellan olika parter. (International Electrotechnical Commission, 1997).

Det finns ett flertal olika dokument som kunden bör förbereda för att specificera systemets funktionalitet. Om dokumenten är gjorda ordentligt kommer det att vara till stor hjälp för tillverkaren att konstruera systemet i enlighet med vad som beställts. Dokument som ofta görs av kunden är:

- specifikationer
- avtal
- funktionskrav
- orsak- och effektdiagram
- instrumentlista, t.ex. TAG – I/O-typ – skalning – enhet
- lista över alarm meddelanden.

(International Electrotechnical Commission, 2006).

Tillverkaren ska förbereda följande dokument:

- systembeskrivning
- systemdiagram och kommunikation (layout)
- manualer och certifikat
- maskinvarulayout
- I/O-lista med tag-namn
- testresultat från förtesten
- program- och programvaruspecifikation
- testplan
- lista på vad som levereras.

(International Electrotechnical Commission, 2006).

Ett lyckat test behöver ett bra och exakt schema. Schemat ska innehålla en lista över aktiviteter som ska göras och ett tidsschema. För att säkerställa ett effektivt test, ska både

tillverkaren och kunden gemensamt godkänna schemat. Ett schema kunde innehålla följande åtgärder:

- introducerande möte (genomgång av dokument, schema, m.m.)
- kontroll av tillverkarens dokumentation
- kontroll av maskin- och programvara
- mekanisk inspektion
- inspektering av kablage
- introducerande test
- systemets allmänna funktionalitet
- visualisering
- komplicerad funktionalitet
- avslutande möte för FAT.

Det introducerande mötet är den mest avgörande delen för genomförande av FAT. Alla dokument ska genomgå och scheman ska göras för att säkerställa att FAT blir utfört under rimlig tid. Den egentliga FAT börjar efter det introducerande mötet. (International Electrotechnical Commission, 2006).

4.2 Testets genomförande

FAT ska utföras av tillverkaren och kunden ska bevittna alla skeden i testet. FAT:en ska omfatta områden som projektrelaterade delar, programrelaterade funktioner, automationssystemets programvara och maskinvara samt systemrelaterade funktioner. Tidigare nämnda åtgärder kommer att bekräfta automationssystemets och det överordnade systemets funktionalitet. Testets åtgärder kan delas in i projektrelaterade delar, systemegenskaper och användargränssnitt. (International Electrotechnical Commission, 2006).

Testutrustning

En stor del av testutrustningen kan man vara tvungen att leverera till anläggningen efter att FAT har utförts. Exempel på sådan utrustning kan vara datorer, switchar, reläer och programvara som är installerad i datorerna, utrustning för det lokala nätverket, modem eller nätverksadapttrar. Vissa test kräver simulatorer för att simulera liknande omgivning

som i verkligheten, sådan apparatur skickas inte till anläggningen. (International Electrotechnical Commission, 2006).

Projektrelaterade delar

Projektrelaterade delar omfattar all dokumentation som är relevant för FAT, maskinvara, programvara och kablage. All utrustning som ska levereras och all dokumentation bör genomgå för att säkerställa att de motsvarar kundens beställning. Efteråt görs en genomgång av all dokumentation som är relevant för FAT. Exempel på dokument som kan vara relevanta för FAT är t.ex. layoutritningar, beställningar, maskinvaruritningar eller projektspecifikationer. Det är viktigt att man efter genomgången av dokumenten gör en kontrollista, som båda parterna godkänner och undertecknar. (International Electrotechnical Commission, 2006).

Maskinvaran kontrolleras för att säkerställa att arkitekturen, antalet, storlekar, m.m. är i enlighet med dokumenten. Även programvarulicenser, reservdelar och konsumtionsvaror ska kontrolleras. Referensdokument används under maskin- och programvarukontrollen. De vanligaste referensdokument är tillverkarens godkända maskinvaruritningar och beställningen där all utrustning är specificerad. (International Electrotechnical Commission, 2006).

Systemen kan bli levererade till olika omständigheter, allt från varmt och fuktigt klimat till kallt och torrt. Därför bör utrustningen vara väl planerad och anpassad till anläggningens förhållanden. För att säkerställa att skåpen och maskinvaran är tillräckligt bra planerade till kommande förhållanden är det skäl att göra en mekanisk kontroll av systemet. Kontrollen börjar med kontroll av kabelgenomgångar, stödbalkar och tillbehör. Vid kabelgenomgångar bör man kontrollera att de är tillräckligt täta och att det inte finns synliga hål. Oftast är kabelgenomgångar som kommer uppifrån genom skåptaket mer sårbara för eventuella läckage än de som kommer från sidorna eller nerifrån. Stödbalkarna bör även granskas, ifall att de inte är tillräckligt bra fastsatta kan de med tiden bli lösare eller t.o.m. lossna och därmed utsätta skåpet för fara. Kontroll av stödbalkar kan göras visuellt och förhand. Tillbehör, såsom kabelklämmare, kan kontrolleras på samma sätt som stödbalkarna. (International Electrotechnical Commission, 2006).

Man gör även en kontroll av kablagen för att säkerställa att kablarna är kopplade och att inga synliga fel finns. Man bör även kontrollera att säkringarna är av rätta storlekar samt att de säkrar rätta kablar. En noggrann kontroll av att kablarna är rätt kopplade är viktigt

för att undvika problem vid inkoppling av ström. (International Electrotechnical Commission, 2006).

Till kontroll av systemegenskaper hör test av systemuppstart, systembelastning, systemalarm och maskinvaruredundans. Systemegenskaperna kontrolleras för att säkerställa att systemet inte har några problem med uppstart eller återställning efter strömavbrott. Systemuppstart kontrolleras för att säkerställa att systemet kommer att starta på rätt sätt efter att det tagits ur bruk. Testet börjar från att all apparatur är avstängd. Efter att systemet sätts på ska det apparaturen starta upp och börja kommunicera automatiskt. Systembelastningen kontrolleras för att säkerställa datorers funktionalitet. Kontrollen omfattar kontroll av kapaciteten på arbetsminnet, lagringsutrymme och övrig nödvändig kontroll för att säkerställa funktionen. (International Electrotechnical Commission, 2006).

Test av systemalarm görs för att säkerställa att information om systemrelaterade fel, skåpalarm och systemgenererade alarm dyker upp i systemet. Lista över alarm meddelanden kan användas som referensdokument. Alarmtest borde innehålla test av strömavbrott, säkringar, fläktkylning, kommunikation och processobjektalarm. (International Electrotechnical Commission, 2006).

Redundanta system kontrolleras eftersom att det är viktigt att säkerställa att systemet inte får tekniska fel och för att se att allting fungerar som det ska. Systemet kan innehålla ett flertal redundanta komponenter: t.ex. redundant kommunikation, strömförsörjning och systemservrar. Av systemservern kan det finnas en redundant systemserver, vilket betyder att det finns två identiska systemservrar där den andra tar över ifall den ena slutar fungera. Detta är ett sätt att säkra funktionaliteten av kritiska komponenter. (International Electrotechnical Commission, 2006).

Ett användargränssnitt består av bilder, dialoger, process, historik och rapport databaser, m.m. Kundens behov och processens funktioner kommer att bestämma hur gränssnittet blir. Test av programvaran sker genom en visuell och funktionell kontroll av HMI-bilder, alarm och kvittering av alarm. Dessa test täcker den grundliga funktionaliteten hos HMI-programmet.

4.3 Slutförande

En kategoriserad lista görs över ofärdigt arbete och eventuella brister i systemet. Listan kontrolleras efter FAT. Exempel på olika kategorier som kan finnas på listan:

- Ska fixas och omtestas under FAT. Testet fortsätter efter ändringarna.
- Pågående ändringar under FAT.
- Ska testas igen.
- Modifieringar som görs efter FAT, innan systemet levereras till anläggningen.
- Kvarlämnande arbete som ändras på anläggningen.

Alla ändringar borde ske under testets gång. Om detta inte är möjligt, borde det ske efter testet. Parterna som medverkar vid FAT ska tillsammans komma överens om ändringar. Under en FAT skapas ofta en massa dokument, där de flesta kräver aktivitet från både tillverkaren och kunden. Alla testdokument bör skrivas ut och undertecknas. Övriga dokument som har genererats under FAT bör dateras och undertecknas. Lista på brister bör genomgå och kontrolleras att alla nödvändiga ändringar har gjorts. FAT:en är färdig när tillverkaren utfört allt enligt FAT-förfarandet och specifikationerna. Efter en helt färdigställd FAT bör en representant från kundens sida och tillverkaren underteckna ett FAT-certifikat.

5 PLC

En PLC är ett styrsystem som är tillverkat för att fungera i en industriell miljö. PLC:n är programmerade för att läsa av, aktivera och styra industriell utrustning och innehåller därför möjligheten till ett stort antal I/O vilka stöder vida områden av elektriska signaler. (Crispin, 1997).

5.1 PLC-arkitektur

Den interna maskin- och programvarukonfiguration av en PLC är vad som omfattas av dess arkitektur. En PLC:s viktigaste delar är, CPU, minne samt in- och utgångar. Dessa element är integrerade kretsar som består av halvledarkomponenter. Dessa är sammankopplade med hjälp av bussar. En bus är en grupp ledningar där digital information överförs parallellt. I de flesta system finns tre olika bussar: databus, adressbus och styrbussen. (Crispin, 1997).

PLC:ns åtgärder bestäms av programvaran som användaren har gjort. Högnivåprogrammering såsom ladder diagram konverteras till binär form så att de kan lagras i ett digitalt minne såsom RAM eller programmerbar ROM. Vardera instruktion körs av CPU:n. (Crispin, 1997).

En CPU är uppbyggd av en mikroprocessor. CPU:ns uppgift är styra användningen av minne, I/O-kort som den är ihopkopplad med och att köra den kod som användaren har gjort. En mikroprocessors viktigaste element är, register, ALU och styrenhet. Ett register är en byte, word eller longword av minne som är en del av mikroprocessorn, i motsats till allmänt minne som inte är ett integrerat minne. Ett register används för temporär lagring av data och adresser inom CPU:n. En ALU utför aritmetiska och logiska operationer, såsom addition och subtraktion, på data i register. Styrenheten består egentligen av flera räknare och logiska grindar som styrs med klockpulser. Dess funktion är att styra enheterna inom mikroprocessorn för att försäkra sig om att operationerna blir utförda i rätt ordning. (Crispin, 1997).

Minnesenheter lagrar digitala siffror på skilda platser med unika adresser. Minnesenheter består av integrerade kretsar vilka har en adress ingång (oftast 16 bitar) och en in/ut dataport (oftast 8 bitar). Det finns två huvudtyper av minne, RAM och ROM.

Största delen av det totala adresserbara minnet är av RAM-typen, från vilken CPU:n kan läsa information från och skriva till. RAM används för att lagra program och data. För RAM-minnet krävs ett skilt batteri för att upprätthålla data i minnet även när det inte finns extern strömförsörjning. PLC:ns operativsystem finns lagrad i ett minne av ROM-typen. När minne av ROM-typen en gång har blivit programmerad kan man endast läsa av minne utan att kunna skriva till det. ROM-minnet förlorar inte dess data fast strömförsörjning saknas. En minneskarta är ofta skapad för att redogöra hur minnet är allokerat. (Crispin, 1997).

Varje in- och utgång till en PLC har en unik adress för att kunna identifieras. IEC 1131-3, standarden för programspråk, föreslår en metod till direkt representation av data som associeras med ingångar, utgångar och minne till en PLC. Denna metod baserar sig på att PLC:ns minne är organiserad i tre områden, ingångsminne (I), utgångsminne (Q) och internt minne (M). Allt minnesutrymme, även de som representerar I/O-bitar kan bli direkt refererade enligt nedanstående:

% (första bokstavskod) (andra bokstavskod) (numeriskt fält)

Där %-tecknet indikerar en direkt refererad variabel. Den första bokstavskoden specificerar minnesområdet:

I = ingångsminne

Q = utgångsminne

M = internt minne

Den andra bokstavskoden specificerar hur minnet är organiserat:

X= bit

B = byte

W = word

D = double word

L = long word

Om en andra bokstav inte är angiven antas det att vara en bit. Det numeriska fältet används för att ange minnespositionen. För att göra det lättare att minnas specifika in- och utgångar, kan de även tilldelas namn såsom 'starta_motor' som mycket tydligare berättar vad signalen har för funktion. (Crispin, 1997)

Det vanligaste sättet som en PLC exekverar programmet på är att cykliskt läsa av förändringar som görs på ingångarna. I exekveringscykeln börjar PLC:n med att läsa av ingångarna och skriver deras status till minnet, sedan körs programmet uppifrån och ner. Till slut skrivs rätta värden till utgångarna i enlighet med programmet. Cyklisk exekveringsmetod har en del begränsningar p.g.a. att PLC:n måste kunna utföra en del saker under en viss tid. Ett PLC-system som tillåter avbrottsrutiner kan användas för att få PLC:n att genast utföra en viss åtgärd. En avbrottsrutin är en speciell signal till CPU:n som gör att den slutar med huvudprogram exekveringen för en stund för att utföra en annan sak däremellan. (Crispin, 1997).

5.2 Programmeringsspråk

I standarden IEC 61131-3 finns nyligen utvecklade programmeringsspråk ämnade för PLC-programmering. Strukturerad text, ladder diagram och funktionsblock diagram är några exempel på programmeringsspråk som behandlas i standarden. (International Electrotechnical Commission, 2003).

Strukturerad text är ett textbaserat språk, liknande som Pascal, för utveckling av väl strukturerad programvara. ST innehåller kommandon såsom IF ... THEN, CASE, FOR, WHILE, REPEAT. En del matematiska funktioner är även definierade i standarden t.ex. SIN, COS, TAN, SQRT, LOG. (International Electrotechnical Commission, 2003).

Ladder-diagram är ett grafiskt programmeringsspråk inspirerat av elektriska scheman som används av nästan alla PLC-tillverkare. Språket i sig själv kan ses som en del kontakter och relän (öppnande och slutande) som är ihopkopplade med en elektrisk pol på vänstra sidan och en med motsatt polaritet på högra sidan. Programmen byggs upp i olika nätverk och om ett nätverk är sammankopplat från vänstra sidan till högra så utförs det som relät styr. (International Electrotechnical Commission, 2003).

Funktionsblocksdiagram är ett grafiskt programmeringsspråk där programmet består av många funktionsblock som är ihopkopplade. Varje funktionsblock syns som rektanglar där ingångarna till blocket finns på vänstra sidan och utgångarna på högra. Blocken sammankopplas så att en utgång från ena blocket kopplas till ingången på följande block. IEC-standardens definierar en liten del av standard funktionsblock. Special funktionsblock såsom PID-regulator och ramp funktioner kan man själv skapa med ST. (International Electrotechnical Commission, 2003).

6 HMI

Grundprinciperna för presentation av information i användargränssnittet är bl.a. informationen ska vara koncis, lättläst och den ska vara placerad på ett logiskt sätt. Man ska kunna sammankoppla eller urskilja olika information och den ska vara synlig och lätt tillgänglig för dess användare.

Systemets användare bör ha tillgång till dialoger och annan relevant information om systemet. Det ska även vara möjligt att snabbt få tillgång till den information som behövs. Det är viktigt att användaren lätt kan hitta information om en specifik uppgift eftersom tvetydig information kan leda till onödig stress eller misstag. Det bör därför finnas en homogen logik mellan programsidor, både på utförandeplanet och på sättet att presentera information. En sådan sammanhållning gör systemet mera förutsägbart för användaren, vilket i sin tur förbättrar medvetenheten och tolkningen av programmet då användaren är bekant med krävd aktivitet. Systemet ska även innehålla lämpliga symboler eller meddelanden så att användaren håller sig medveten om vad som sker. Användaren ska inte behöva memorera information under en längre tid utan informationen ska anpassas till varje interaktion mellan systemet och användaren. Användaren ska känna att den hela tiden har full kontroll över systemet. Systemet bör ge respons på utförd aktivitet för att användaren ska vara medveten om situationen. (Moller, 2007).

6.1 Färganvändning

Färger spelar en stor roll då man utvecklar användargränssnitt på grund av att de kan förbättra användarens medvetenhet och systemet då färgerna användas på rätt sätt. Då man utvecklar ett användargränssnitt måste man beakta användare med specialbehov, såsom färgblindhet. Färgblindhet betyder inte att man inte kan se färger överhuvudtaget utan det handlar ofta om att man inte kan skilja på olika färger såsom t.ex. röd och grön. En person med röd/grön-färgblindhet kan t.ex. inte se skillnad mellan blå och lila, eller grön och khaki. Sådant problem kan lösas t.ex. genom att man använder både text och färg i användargränssnittet. (Johnson, 2010).

Vår förmåga att upptäcka skillnader i färger är begränsat. Olika våglängder ger upphov till perception av en viss färg, de färger som saknar en motsvarande våglängd kan vi inte se. Faktorer som påverkar vår förmåga att urskilja färgerna från varandra är deras nyans,

styrka/mättnad och klarhet. I allmänhet anses det att färgerna har väldigt liknande valör, dvs. ljus- mörkhetsgrad oberoende av vilken nyans man talar om. Det kan t.ex. vara svårt för oss att se skillnaden mellan två ljusa färger av olika nyanser, desto ljusare dessa färger blir desto svårare är det för oss att skilja dem åt. Klara färger påverkar på mindre ytor medan mindre klara färger påverkar på större ytor. Detta är bra att komma ihåg om man använder text i ett användargränssnitt. Smalare och mindre färgade objekt är det svårare att skillnader på eftersom att en text ofta är tunna sträck. Om avståndet mellan olika texter är långt kan det vara sårare att utskilja deras nyans, eftersom att ögonen måste göra en större rörelse för att kunna fokusera sig på dem. (Rihlama, 1997; Luukkainen, Raivio & Väliaho, 2007).

6.2 Konsekvent design

Programmet bör allmänt sett ha en konsekvent design. Konsekvent användning av olika funktioner gör det lättare för programmets användare att använda programmet, men även för dess utvecklare när denne inte bör tänka på var saker ska placeras. Då man använder konsekvent design på rätt sätt förbättras programupplevelsen för användaren. Konsekvent design bör göras med ett syfte. Att göra ett utseende konsekvent är dumt ifall att det inte skapar mervärde för användaren. I vissa fall måste man välja mellan att göra programmet konsekvent eller användarvänligt p.g.a. att dessa inte alltid går hand i hand. I vissa fall kan konsekvent design göra användarupplevelsen sämre.

Då man designar ett användargränssnitt ska man försöka minimera saker som användaren måste lära sig. Människor gillar i allmänhet inte att lära sig nya saker och ifall de tar sig tid att lära sig någonting, så förväntar de sig att de ska kunna använda den inlärd kunskapen vid flera tillfällen. Därför ska man återanvända funktionsdugliga, existerande, funktioner och koncept om inte syftet med förändringen är att förändra ett beteende eller en användarfunktion. Om vissa funktioner i programmet inte fungerar som förväntat, ska man till att börja med försöka utveckla dessa funktioner så att de lämpar sig och uppfyller sitt syfte. Om detta inte lyckas utvecklas en helt ny design. Då det görs förändringar i designen måste man komma ihåg att tillämpa dem igenom hela programmet. Om man är tvungen att använda sig av specialfall, lokal optimering av en funktion som inte används globalt, ska man se till att den bästa möjliga optimering utförs. Dessa tumregler hjälper programdesignern att hålla systemet så konsekvent som möjligt. (Bercun, 1999).

7 TIA Portal

TIA (Totally Integrated Automation) Portal är ett verktyg för utveckling av programvara till Siemens logiker. TIA Portal är den senaste versionen (V11) av verktyget. I verktyget är STEP 7 och WinCC integrerat som ett enda verktyg. Av programmet finns det olika versioner, funktionalitet och pris skiljer dem åt. I STEP 7 finns versionerna basic och professional samt i WinCC finns basic, comfort, advanced och professional. Fördelarna med att ha ett gemensamt verktyg för logik- och användargränssnitts-programmering är många, t.ex är det enklare att hantera taggar mellan dessa delar. Det är på flera ställen också möjligt att använda sig av drag & drop-funktion när man ska flytta olika taggar. Vid import av större datamängder kan Microsoft Excel användas. Funktioner, egenskaper och bibliotek visas automatiskt på det mest relevanta sätt för pågående aktivitet. Det är väldigt enkelt att byta mellan programmeringseditorerna då det finns drag&drop-funktioner som fungerar på ett intelligent sätt. (Siemens, 2011a, Siemens, 2011c). Programutvecklaren kan välja mellan två vyer, ”portal view” som är en uppgifts-orienterad layout som är grafiskt och enkelt samt ”project view” som gör att man har snabbt tillgång till alla behövliga verktyg (Siemens, 2011c).

Styrkan med TIA Portal är att dess komponenter finns i samma verktyg, därmed har de samma utseende, vilket gör det enkelt när man vill få olika komponenter att kommunicera med varandra. Delade funktioner såsom nerladdning, integrerad krossreferens och onlinefunktioner kan enkelt manövreras från samma programram. I TIA Portalen finns det två typer av bibliotek, projektbibliotek och globala bibliotek. Bibliotek används för att man ska kunna spara ett element eller objekt som man har konfigurerat för ett specifikt behov. Detta är för att man enkelt ska kunna återanvända element för att spara tid och effektivisera arbetet. I projektbiblioteket finns sådana objekt som ofta behövs eller används i det pågående projektet. Projektbiblioteket öppnas, stängs och sparas i takt med projektet. Globala bibliotek är bibliotek som sparas som skilda filer och därmed kan öppnas och editeras från olika projekt. Biblioteken kan innehålla två typer av objekt, sådana med centralt modifierade egenskaper och konfigureringar, samt mallar som kan bestå av en grupp med objekt. Dessa kallas för en faceplate. (Siemens, 2011c).

WinCC är ett programmeringsverktyg med vilken man kan skapa användargränssnitt till PLC-programmen som man skapar med STEP7. Programmet har lätta grafiska funktioner med vilka det är lätt att skapa dynamik i processobjekten, med t.ex färgförändringar. För

mer komplicerade dynamiska funktioner finns det möjligt att implementera VB eller C skript. I WinCC finns många fördefinierade funktioner för interaktiva objekt, såsom knappar och brytare. För att skapa grafiskt snygga bilder finns det t.ex. färdigt ritade rör som med skuggning skapar en tredimensionell illusion. Ifall att programvaran börjar utvecklas innan maskinvaran är anskaffad finns det möjligheten att simulera kompletta system. Med ett par inställningar kan man snabbt testa WinCC-programvaran och ifall PLC-simulationen är startad kan man även testa kommunikationen med den. Faceplate editorn är ett verktyg som används i WinCC. Med hjälp av detta verktyg kan man skapa en mall, en sk. Faceplate. I denna mall infogas objekt, t.ex. knappar och brytare, vilka kan användas för exempelvis motorstyrning. Ofta kopieras mallen så att det finns flera olika instanser av faceplaten i programmet. Då man gör en förändring i mallen med hjälp av faceplate editorn förändras denna information i alla instanser av faceplaten. Detta gör programmeringen enklare, effektivare och man sparar mycket tid. (Siemens, 2011c).

STEP 7 är ett programmeringsverktyg för logikprogrammering. Utseendet har förändrats mycket från de tidigare versionerna av STEP 7. En förbättring som har gjorts är att taggar som ändras på ett ställe förändras nu överallt i programmet, t.ex om man ändrar tag-namnet i tag-listan så ändras det på samtliga ställen i programmet automatiskt. Denna egenskap är global så detta gäller även för HMI-taggar. De tillgängliga programmeringsspråken är samma som tas upp i IEC 61131-3. (Siemens, 2011b).

8 Testsystemets förverkligande

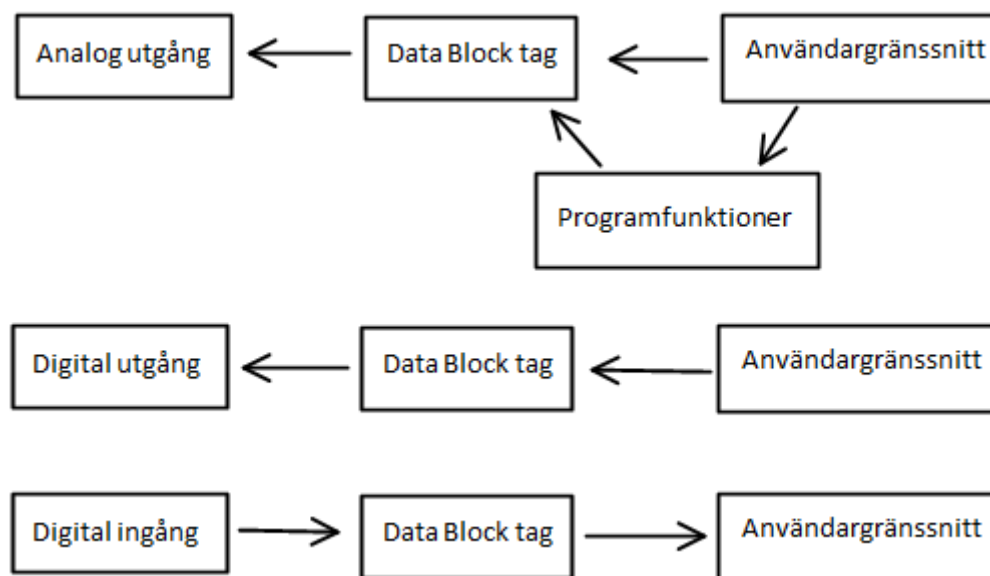
Projektet delades upp i två separata examensarbeten. Det ena arbetet omfattar planering av maskinvaran och ritning av elschema medan detta arbete omfattar planering av programvaran samt programmering av PLC och användargränssnitt. Dessa två arbeten är oberoende av varandra även om de är väldigt nära relaterade till varandra.

Testsystemet byggs in i ett skåp som är försett med hjul så att man lätt kan flytta det. Systemet består av Siemens S7-300 seriens logik, 64 digitala ingångar, 128 digitala utgångar, 72 analoga utgångar samt två ET200 kommunikationsmoduler. Med detta system ska man kunna simulera ett kraftverks olika givare där man kan med de analoga utgångarna simulera t.ex temperaturförändringar. Detta är för att man ska kunna testa säkerhetsfunktionerna på det testade systemet. För att lättare kunna kontrollera testet i efterhand loggas allt data och man kan även se grafer på de kanaler man vill. Ifall testaren vill göra specialfunktioner för specifika projekt kan sådana skapas och användas. Testsystemet kopplas till det testade systemet in med en specialkabel. I ena ändan kopplas testsystemet med en snabbkontakt och i andra ändan kopplas de enskilda kablarna till dess rätta radklämmare.

8.1 Planering och programmering

Grunden för kravspecifikationen var ett protokoll från ett grundande möte. Dessa krav blev noggrannare specificerade under hösten på ett annat möte samt efter ett antal diskussioner med blivande användare. Utifrån dessa samtal planeras hur PLC-programmen borde fungera samt ett utkast på hur användargränssnittet kunde utformas. Med detta som grund började programmeringen.

Programmeringsspråket som programmet är uppbyggd av är FBD, som redan tidigare nämnts är FBD ett av de vanligaste programmeringsspråken. FBD väljs för att det är lättast att göra den här typen av program med det språket. Värden på in- och utgångarna lagras i datablock. Programfunktionerna som aktiveras från användargränssnittet beräknas och sedan lagras sedan i datablocken. Blockschema för programmets uppbyggnad syns i figur 2.



Figur 2. Blockschema över programuppbyggnaden.

8.1.1 Maskinvarukonfiguration

Maskinvaran konfigureras så att det i rack 1 placeras CPU:n samt ethernetkortet CP 343-1. Som CPU används en S7-317F, som är en säkerhetslogik. Det finns ingen egentlig teknisk orsak till att en säkerhetslogik borde användas, utan uppdragsgivaren hade en överlopps sådan och därför tas den i bruk. Kommunikationen till rack 2 och 3 görs med profibus där CPU:ns profibusportar kopplas till ett IM 153-2 kort i de två andra racken. I rack 2 och 3 sätts alla I/O-kort.

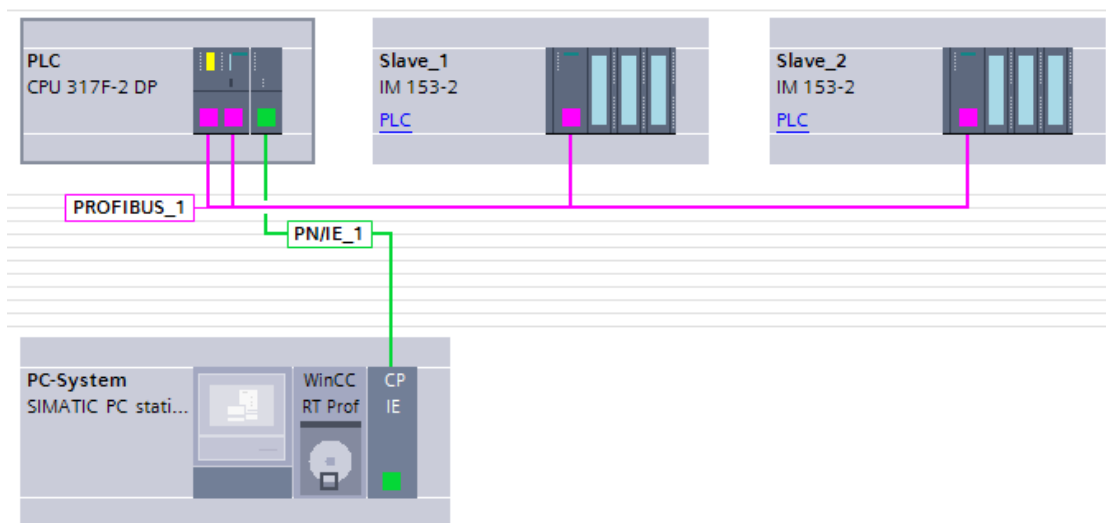
Rack 2:

- 2 st DIx32
- 4 st DOx32
- 2 st AOx8.

Rack 3:

- 7 st AOx8

Kommunikationskorten placeras i tre olika rack för att de inte får plats i samma. En CPU används. En översikt på maskinvarukonfigurationen kan ses i figur 3.



Figur 3. Översikt på maskinvarukonfigurationen.

Som det framgår ur figur 2 ska PC:n som användargränssnittet görs på konfigureras i samma maskinvarukonfiguration. PC:n kopplas ihop med PLC:n via en ethernet kabel. På grund av att PLC:n inte har någon ethernetport används ett externt ethernetkort (CP 343-1)

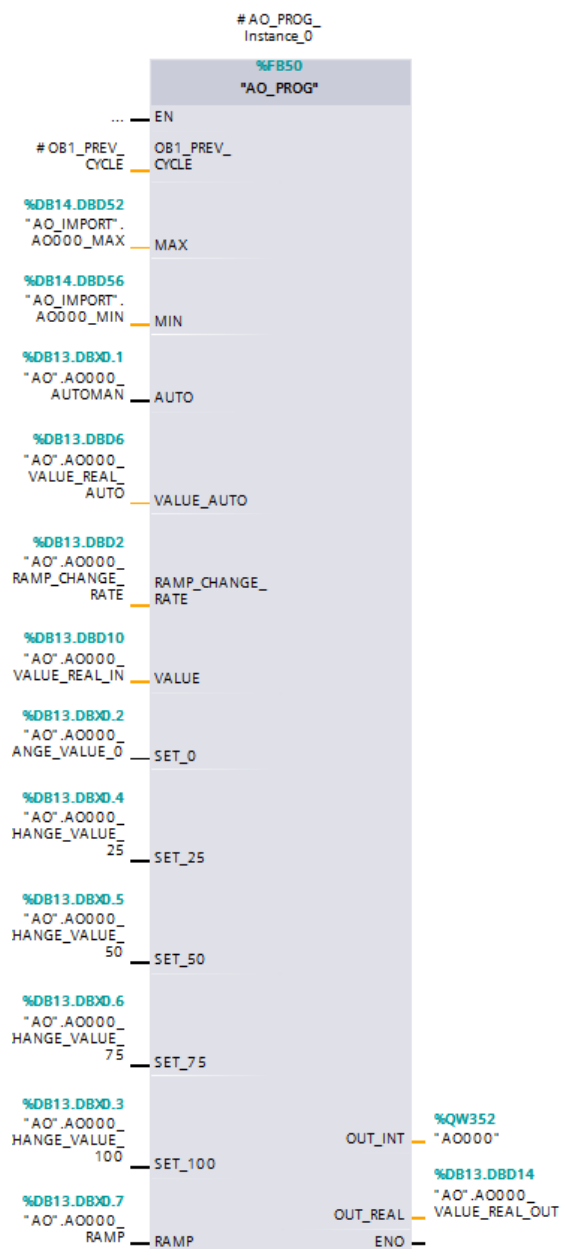
8.1.2 PLC

För samtliga olika I/O-typer (DI,DO,AO) skapas olika grupper. För dessa grupper delas kanalerna in så att 8–10 kanaler används per programblock. Programblocken som används för I/O-programmen är funktionsblock. FB används främst till att temporärt lagra data på de taggar som använts i ett datablock och kan därför användas från andra programblock efter exekvering av blocket. Alternativet till FB skulle vara FC där samma sak exekveras, men där sparas inte taggarnas värden efter exekvering. Detta medför att man inte kan använda sig av föregående värde på taggen i programmet, utan man måste räkna ut ett nytt värde för att använda taggen.

Programmet byggs upp i olika programblock, en för varje kanaltyp. Programblocken innehåller all den funktionalitet som behövs för respektive kanaltyp. Programblocken används så många gånger det finns kanaler. Programmen byggs in i programblock för att det är lättare att i efterhand göra ändringar i programmet, eftersom att förändringar sker på alla ställen blocket använts. Från de tre programblocken kommer de viktigaste delarna att förklaras. Programblocket för analog utgång framgår ur figur 4.

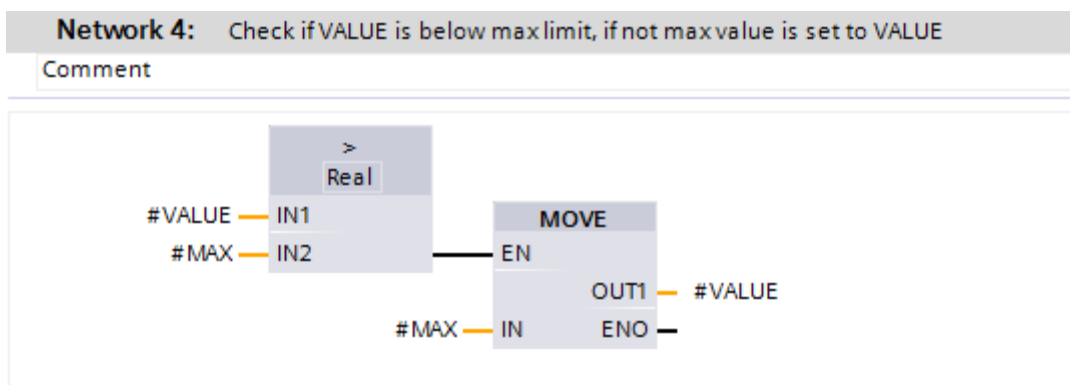
Analog utgång

För att göra det lättare att i efterhand ändra på programmet bör ett programblock med all funktionalitet skapas. Detta block används sedan många gånger, men det används med olika taggar. Innehållet i programblocket som ses i figur 4 och behandlas i detta kapitel.



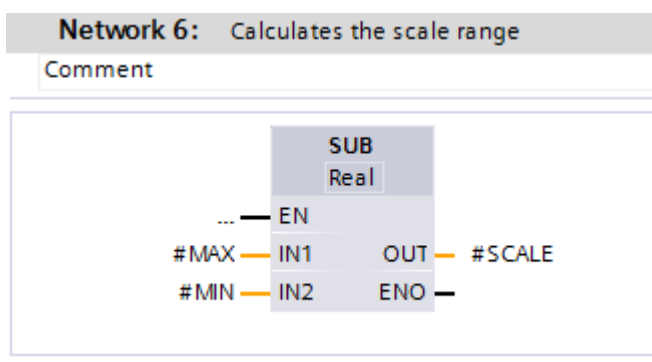
Figur 4. Programblock för analog utgång.

En kontroll görs så att värdet på utgången inte stiger över den användardefinierade maxgränsen, detta görs i nätverk 4. Där jämförs det om värdet ut är större än max-värdet, om värdet är större körs max-värdet som värde ut (figur 5). I nätverk 5 görs en liknande jämförelse, men där jämförs det om värdet understiger minimivärdet.



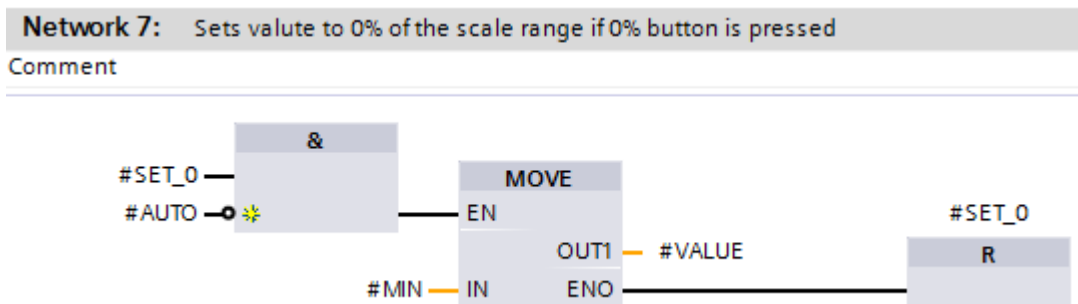
Figur 5. Jämförelse ifall värde ut är större än max-värde.

För att senare i programmet kunna ha nytta av storleken på mätområdet, beräknas detta i nätverk 6. Det görs genom att ta max-värdet och subtrahera bort minimivärdet, vilket sparas i en intern variabel (figur 6).



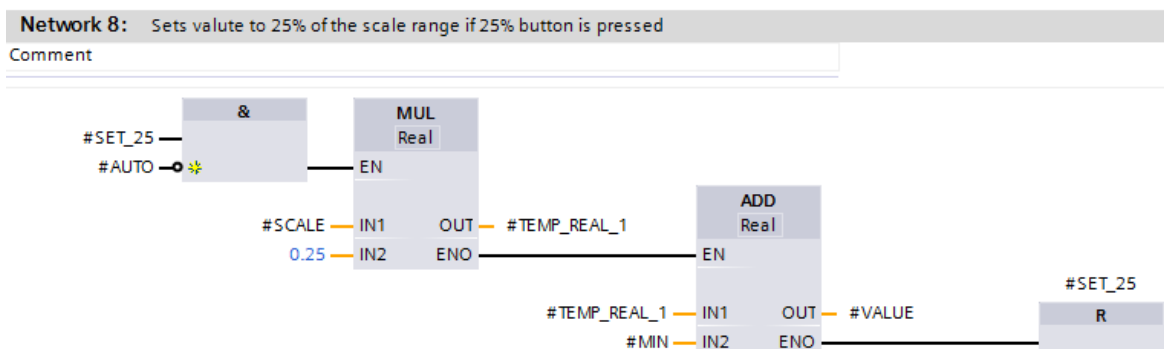
Figur 6. Beräkning av mätområde.

För att användaren snabbt ska kunna testa lineariteten ska det finnas möjligheten att ställa utgången till 0 %, 25 %, 50 %, 75 % och 100 % av mätområdesvärdet. För att ställa utgången till 0 % eller 100 % används min- och max-värden. Värdet ska sättas till min-värde ifall man valt manuellt läge samt att man genom användargränssnittet trycker på 0 %-knappen. För att det ska vid rätt tillfälle nollställa biten som tillåter funktionen att köra, görs detta som sista del i nätverket. Funktionen där max-värde ställs till utgången fungerar på samma sätt, endast min har ersatts med max (figur 7).



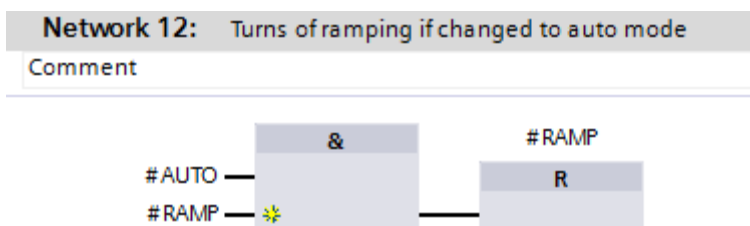
Figur 7. Värdet på utgången sätts till minimivärdet.

Villkoren för att 25 % av värdet på området sätts till utgången är samma, dvs. knappen för 25 % är aktiverad och man har valt manuellt läge. Beräkningen sker så att mätområdet som beräknades i nätverk 6 multipliceras med 0,25, varefter minimivärdet adderas till resultatet. För att det ska vid rätt tillfälle nollställa biten som tillåter funktionen att köra, görs detta som sista del i nätverket. Nätverken där värdet sätts till 50 % och 75 % av mätområdet är likadana med den skillnaden att villkorsbiten samt multiplikationsfaktorn är den korrekta för respektive funktion (figur 8).



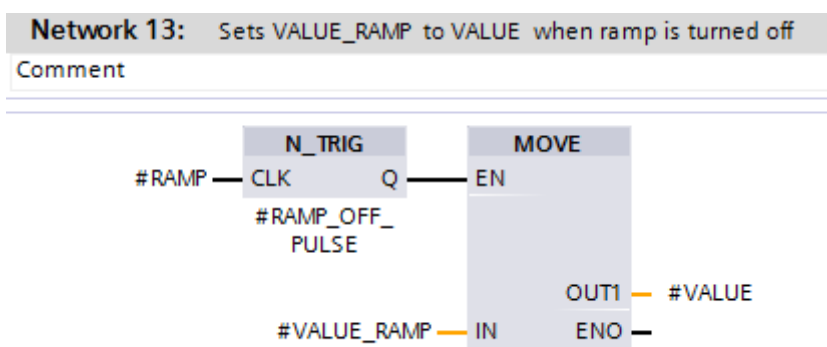
Figur 8. Sätter värdet ut till 25 % av mätområdet när villkoren är uppfyllda.

När man valt automatläge ska man inte kunna köra ramp-funktionen, därför görs en låsning i nätverk 12. Låsningen nollställer ramp-biten ifall man valt automatläge, men då måste rampen vara igång (figur 9).



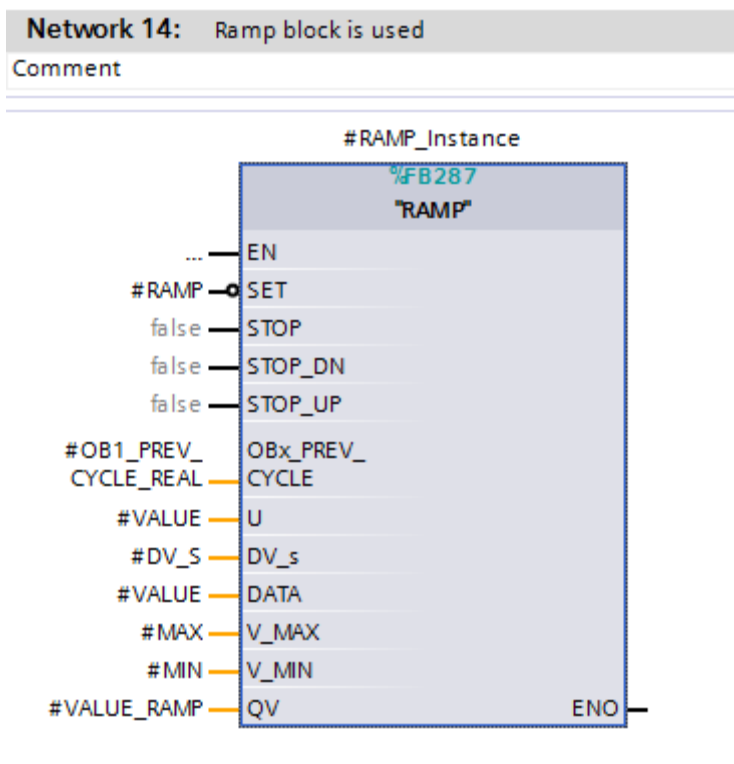
Figur 9. Stannar rampfunktionen ifall den är igång när kanalen sätts i auto-läge.

För att man ska få rampens börvärde att bli lika med rampens värde när rampen stängs av görs nätverket som framgår ur figur 10. Villkoret som utför förflyttningen är när rampens on/off-bit får en negativ flank. Detta nätverk görs för att man ska kunna avläsa vid vilket värde exempelvis något säkerhetsskydd utlöstes.



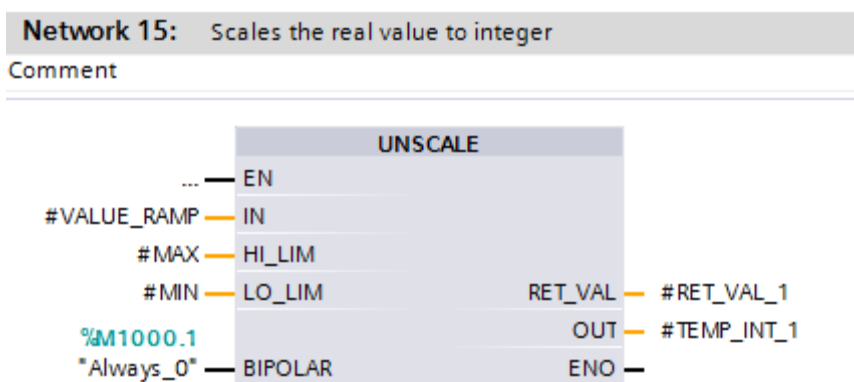
Figur 10. Värdet på rampfunktionen sätts till ingångsvärdet när rampen stängs av.

Ramp-funktionen är ett block som lånades från Vaasa Engineering. Funktionen är skapad i ST-språket. Värdet på U är börvärdet för rampfunktionen, till det strävar utvärdet QV. Så att om U-värdet är lägre än QV börjar värdet på QV minska linjärt och tvärtom. Hur snabbt rampen ändrar värde anges i DV_S där det ursprungliga värdet anges som enheter per sekund. I detta arbete har det ändå ändrats till enheter per minut för att användarna ska få bättre översikt på ett lämpligt värde. För att ramp-funktionen ska kunna köras får inte SET-ingången vara hög. Om SET är hög tvingar det värdet på DATA till utgången QV utan att starta rampen. Det finns begränsningar på hur högt rampen kan rampa i form av min- och max-ingångar, när något av dessa värden överskrids stoppas rampen genast (figur 11).



Figur 11. Rampblocket lånat av Vaasa Engineering Oy.

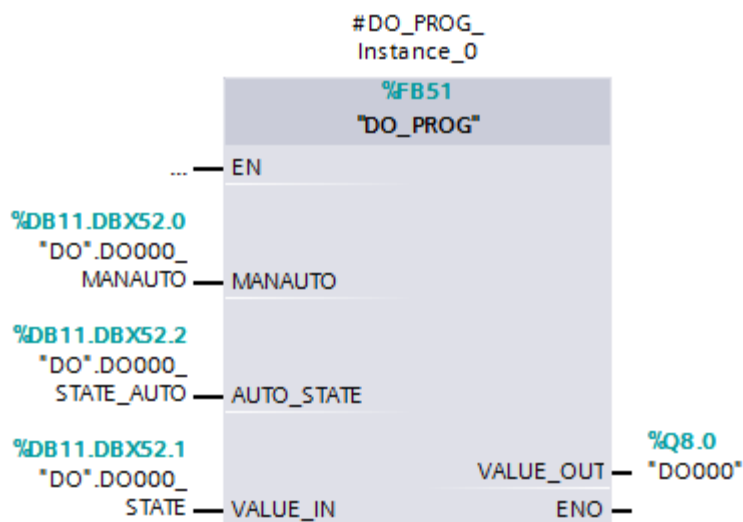
Fysiska utgångar är av datatypen integer, men samtliga datapunkter som programmet jobbar med är av double-typen. För att det ska vara möjligt att skicka ut värdet på den fysiska utgången måste värdet konverteras från double till integer. Detta sker i ett unscale-block som är ett standardblock i programmet. Eftersom vi vill att värdet ska vara mellan 0 och 65 535, ger blocket oss att få värdet bipolärt, dvs. från -32 768 till 32 767. Eftersom värdet ut inte ska vara bipolärt är vi tvungna att sätta BIPOLAR-ingången till blocket att vara en nolla (figur 12).



Figur 12. Värdet på rampen konverteras från real till integer.

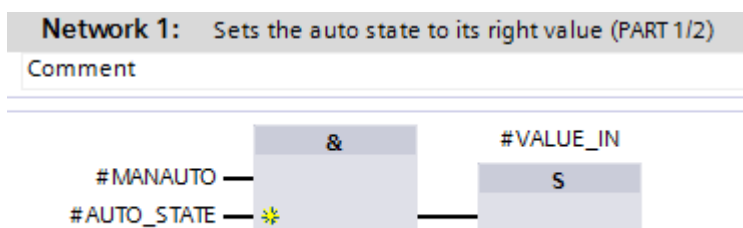
Digital utgång

På samma sätt som i analoga utgångar skapas även i digitala utgångarna ett programblock som används många gånger med olika taggar. Innehållet i detta programblock som ses i figur 13 behandlas senare i detta kapitel.

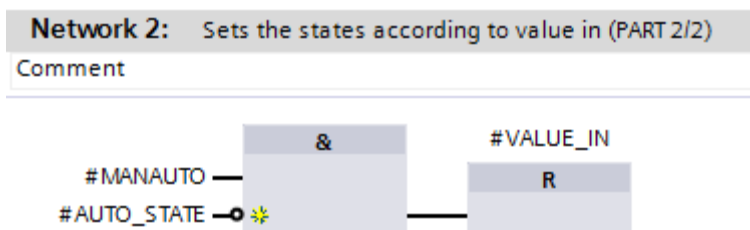


Figur 13. Programblock för digital utgång.

Status-biten är den bit som PLC:n ger ut på utgången. Värdet på de digitala utgångarna kan sättas hög eller låg vid manuellt läge. Automatläget fungerar på samma sätt som i analoga utgångarna, dvs. användaren kan själv göra sin projektspecifika kod som körs. Ifall automatläget väljs kan användaren med sitt program skriva till auto-biten, vilket i sin tur sätter status-biten hög eller låg oberoende på status-biten (figur 14 och figur 15).

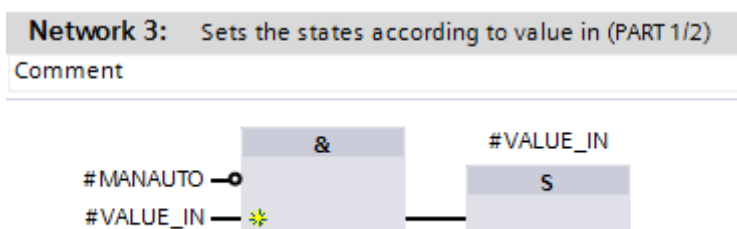


Figur 14. Sätter värdet från auto-biten till status-biten.

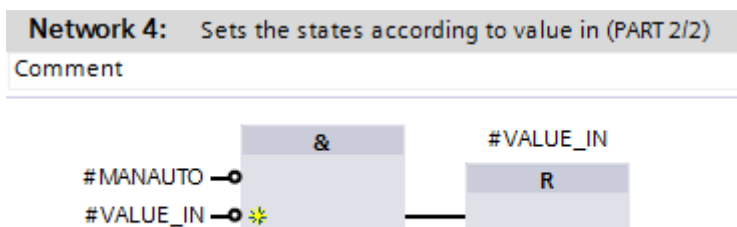


Figur 15. Sätter värdet från auto-biten till status-biten.

När automatläge har valts är det inte längre möjligt att via användargränssnittet ändra status på status-biten. Detta har gjorts på samma sätt som ovanstående nätverk, vilket gör att endast när manuellt läge är valt är det möjligt att ändra värde från användargränssnittet (figur 16 och figur 17).



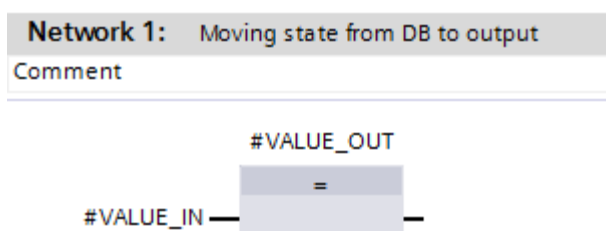
Figur 16. Sätter rätt värde på status-biten.



Figur 17. Sätter rätt värde på status-biten.

Digital ingång

Digitala ingången för endast vidare informationen från fysiska ingången till motsvarande DB-tag, därifrån det sedan läses in till användargränssnittet (figur 18).



Figur 18. Status från digital ingång förs över till motsvarande DB-tag.

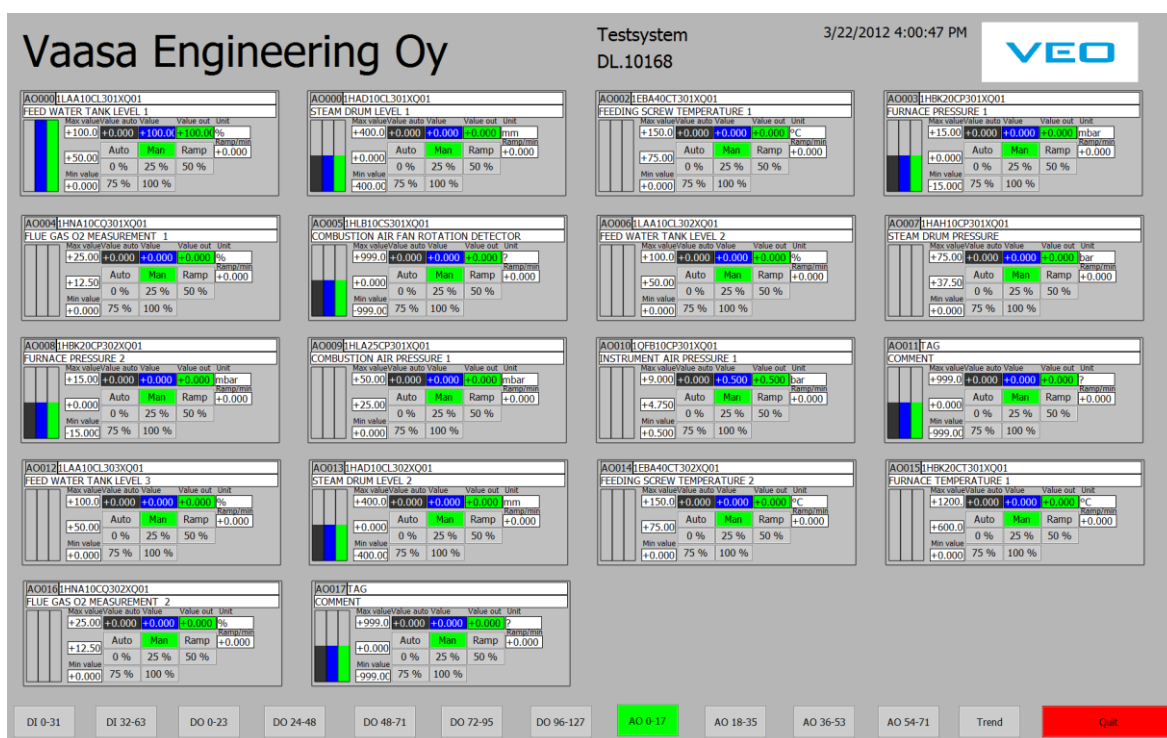
8.1.3 WinCC

För att styra PLC-programmen bör ett användargränssnitt skapas. Som det tidigare nämntes finns verktyget för detta integrerat i TIA Portalen. Ett stort antal likadana in- och utgångar gör att det kommer att finnas flera likadana delar i användargränssnittet. Det finns många fördelar med att förverkliga det med "faceplate". Som det i tidigare kapitel nämntes är en faceplate en modell som kan användas om många gånger. Det skapas skilda faceplates för digitala ingångar, digitala utgångar och analoga utgångar.

Tack vare att man skapar skilda faceplatar för dessa är det lättare att i efterhand ändra på eller förbättra utseendet på dem. Eftersom att informationen kopieras till alla enskilda instanser får användargränssnittet automatiskt en konsekvent design. I samtliga instanser finns högst upp det namn som taggen har och under den finns en beskrivning på vad in- eller utgången representerar.

Analoga utgångar

De analoga utgångarna placeras så att det finns 18 st per sida, eftersom att det på det testade systemets säkerhetslogikkort finns sex kanaler för varje kort. En helhetsbild över layouten för analoga utgångar kan ses i figur 19.



Figur 19. Layout över analoga utgångar.

Användarvänligheten är en central del i utformning av användargränssnitt. Användaren kan välja mellan två lägen, manuell och automat. I manuellt läge kan man använda sig av de funktioner som visades i det tidigare avsnittet. I automatläget kan användaren själv programmera någon projektspecifik funktion. Eftersom uppdragsgivarens kunder också är närvarande under testerna ville man även få fram grafiska förändringar i värden. De grafiska förändringarna visas så att fyllnadsgraden av färg i rektanglar förändras i takt med förändringarna i värden på de analoga utgångarna. Tre olika värden visas i varje faceplate, in-värde, ut-värde och auto-värde. In-värdet är det värde som skickas in till PLC-funktionerna som bör-värde, dvs. när man vill rampa ett värde så visas börvärdet på in-värde som en stegförändring. Ut-värdet är det värde som PLC:n ger ut på den fysiska utgången men i någon ingenjörsenhet. Autovärdet är det värde som är börvärdet när man kör i autoläge.

Som framgår i figur 20 har de tre olika värden olika färger, för att visa vilka färger och vilka rektanglar som hör ihop har de samma färger. För att få helt vanliga rektangelobjekt att fungera som nivåindikatorer finns en funktion i WinCC som heter relativ fyllnadsgrad. Med den funktionen kan man få vilka objekt som helst att visa en dynamisk färgförändring. Skallningen är gjord så att det visar en nivå mellan min-värde och max-värde. Skillnaden visas som ett tal mellan 0 och 100 %.

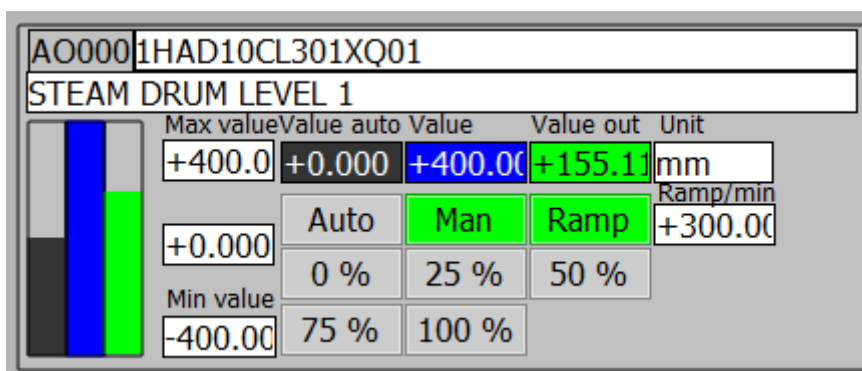
Skalningen görs i ett VB skript där nivån beräknas enligt följande

$$\frac{Max - Min}{Är - Min} = \text{Relativ fyllnadsgrad}$$

Så att man snabbt kan testa lineariteten tillämpas procentknappar. Knapparna ger ett utvärde som är:

$$\text{Procenttal} \cdot (Max - Min) = \text{Utvärde}$$

De möjliga procenttalen som används är 0 %, 25 %, 50 %, 75 % och 100 %. Onödig information lämnas bort, därför visas inte procentknapparna eller ramphastighet när man har i automatläge.

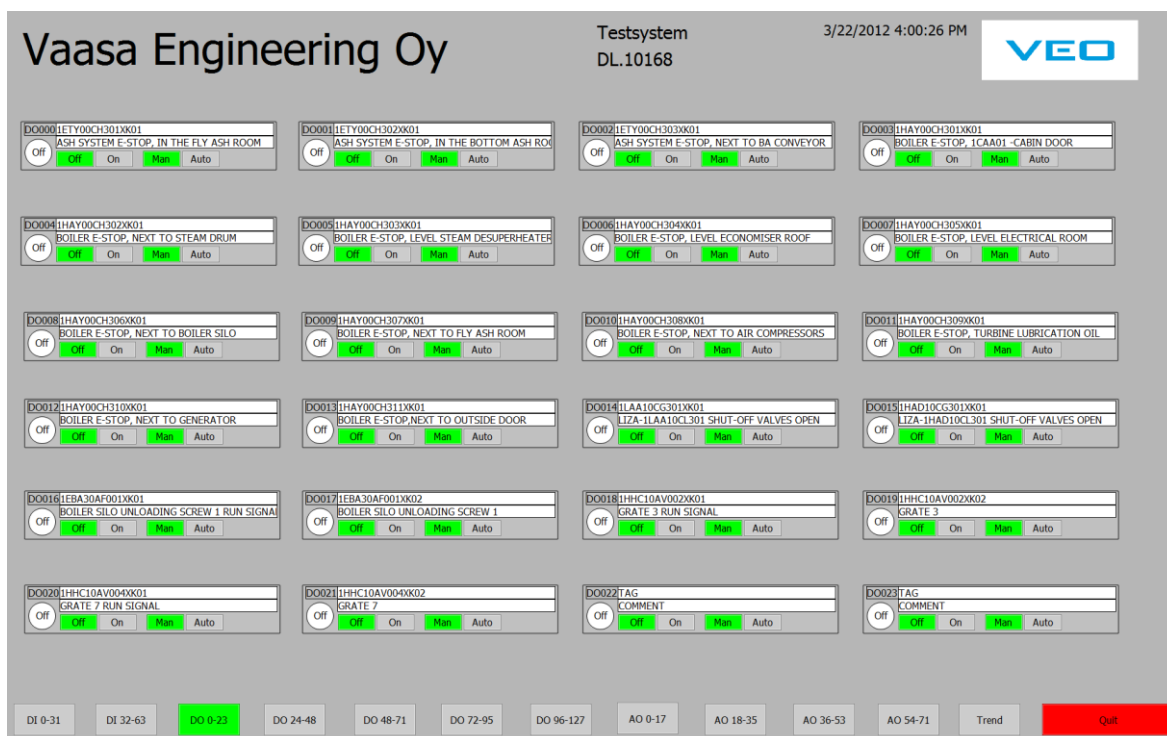


Figur 20. Användargränssnitt för analog utgång.

För att användaren ska vara medveten om vad systemet gör bör användaren se förändringar och få respons från systemet. Responsen förverkligas genom att visa vilket läge man som är aktiverad, t.ex. blir manuell- och automatknapparna gröna beroende på vilket läge användaren valt. Rampknappen blir också grön då rampfunktionen aktiveras. Mätområdets storlek kan variera kraftigt beroende på instrumentet det simulerar, därför är det möjligt att välja ramphastigheten på lämpligt sätt.

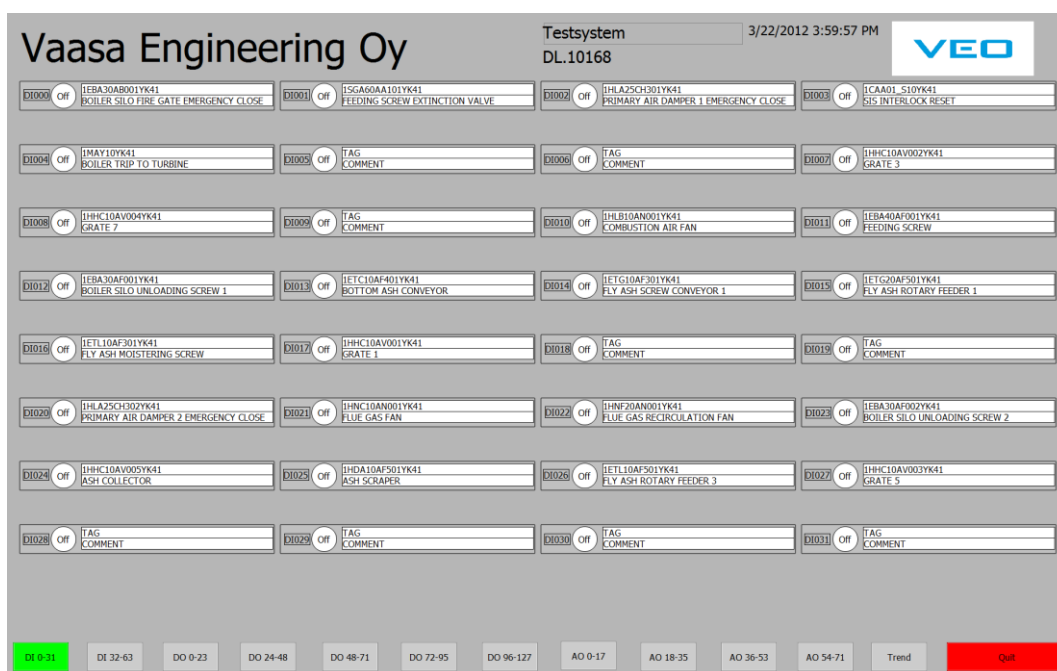
Digitala in-/utgångar

Det placeras 24 st digitala utgångar per sida, eftersom att det finns 12 kanaler per kort på det testade systemets säkerhetskort. En helhetsbild över layouten för digitala utgångar kan ses i figur 21.



Figur 21. Layout över digitala utgångar.

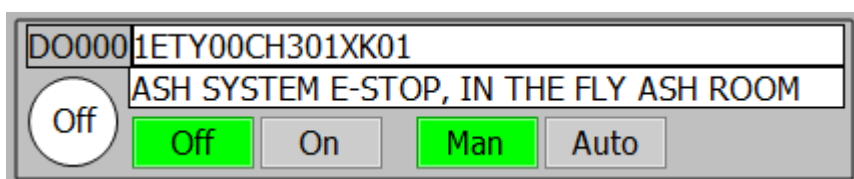
Det placeras 32 digitala ingångar per sida, eftersom det finns 16 kanaler per kort på det testade systemets säkerhetskort. Digitala ingångar placeras 32 st/sida, eftersom på det testade systemets säkerhetslogikkort finns 16 kanaler för varje kort. En helhetsbild över layouten för digitala ingångar kan ses ur figur 22



Figur 22. Layout över digitala ingångar.

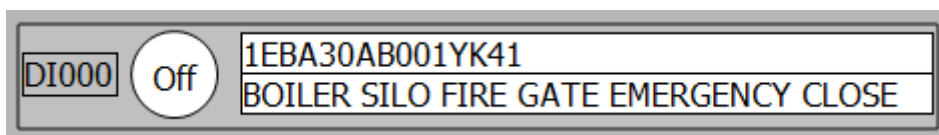
Liksom i de analoga utgångarna kan man också i digitala utgångarna välja mellan manuellt- eller automatläge. Dessutom finns det givetvis möjligheten att sätta utgången hög eller låg med on/off-knappar. För att öka användarens medvetenhet visas det även med färger vilket läge utgången är, man/auto och on/off.

Så det ska tydligt synas, även längre ifrån sett, vilket läge in-/utgången är i. Statusen visas så att cirkeln är grön när den är hög och vit när den är låg. Med tanke på färgblindhet visas statusen inte enbart med en färgförändring, utan det finns även en text-ruta i mitten av cirkeln som visar statusen med en on/off-text (figur 23).



Figur 23. Användargränssnitt för digital utgång.

Skillnaden mellan ingången och utgången är endast att ingången inte går att styra från användargränssnittet, utan statusen läses av ingången och visas i cirkeln. Likadant som i utgången finns även här en text som indikerar ingångens status (figur 24).



Figur 24. Användargränssnitt för digital ingång.

8.2 Importverktyg

Ett importverktyg som importerar data till TIA Portalen var ett krav av uppdragsgivaren för att det ska vara effektivt att påbörja ett nytt test. All data som ska importeras finns redan färdigt till varje enskilt projekt i en Excel-tabell. Eftersom detta dokument skapas till varje projekt ger det möjligheten att kopiera data därifrån. Data som importeras kopieras till en specifik kolumn i ett datablock i TIA Portalen. Detta gör att man vill ha data som ska importeras i en kolumn. Ordningen bör också vara den samma som taggarna i TIA Portalen för att rätt data kommer på rätt plats. När användaren har kopierat en viss del av det tidigare nämnda dokumentet till ett skilt blad i verktygets Excel-boken och angett korrekt ID för den första kanalen, sköter verktyget om resten. Verktyget letar upp data från det inklistrade fältet med ett Excel-kommando med namnet VLOOKUP. Som framgår i

figur 25 är ”EW800” den första analoga ingångens ID till det testade säkerhetssystemet. Verktøget adderar sedan till två för varje kanal. Oftast går inte en typs kanal med löpande numrering genom samtliga kanaler. När mellanrummet inträffar skriver användaren nästa använda ID och därefter fortsätter numreringen.

TAG #	BD tag	Data type	Offset	Start value
EW800	1 AO000_COMMENT	String[50]	0.0	'FEED WATER TANK LEVEL 1 '
EW800	1 AO000_MAX	Real	52.0	100.0
EW800	1 AO000_MIN	Real	56.0	0.0
EW800	1 AO000_TAG	String[20]	60.0	'1LAA10CL301XQ01'
EW800	1 AO000_UNIT	String[10]	82.0	'%'
EW802	1 AO001_COMMENT	String[50]	94.0	'STEAM DRUM LEVEL 1 '
EW802	1 AO001_MAX	Real	146.0	400.0
EW802	1 AO001_MIN	Real	150.0	-400.0
EW802	1 AO001_TAG	String[20]	154.0	'1HAD10CL301XQ01'
EW802	1 AO001_UNIT	String[10]	176.0	'mm'

Figur 25. Importverktyg analog utgång.

Data som ska importeras är:

Digital in-/utgång:

- Tag-namn
- Beskrivning

Analog utgång:

- Tag-namn
- Beskrivning
- Max-värde
- Min-värde
- Enhet

I importverktygets digitala del var man tvungen att göra en liten förändring vid additionen av kanalnummern. I digitala kanaler kan man använda bitarna från noll till sju. Vid bitarna noll till sex ska en bit adderas till kanal ID:n och vid bit sju ska tre bitar adderas till kanal-ID:n. En IF-sats kontrollerar ifall bit-nummern är nummer sju, ifall detta är sant adderas 3 bitar till föregående värde annars adderas bara en bit.

9 Resultat och tolkning

Uppdragsgivaren var i behov av ett nytt och effektivare testsystem för test av säkerhetsautomations system. Ett nytt testsystem behövdes på grund av praktiska skäl, tidigare var man tvungen att skriva upp kanalernas namn på tejpbitar som sedan limmades fast ovanför brytarna och potentiometrarna. Detta förarbete var mycket tidskrävande. Uppdragsgivarens kunder ville även att testsystemet skulle vara mer överskådligt.

Vid fabrikstester medverkar både leverantören och kunden. Kunden reser ofta långt för att delta vid testen, därför bör man sträva till att testen utförs effektivt och snabbt. Noggranna förberedelser och bra testutrustning ger förutsättningar för att testen löper smidigt. Det tidigare testsystemet kan ses i figur 26.



Figur 26. Uppdragsgivarens tidigare testsystem.

Det nya testsystemet är ett användarvänligt och effektivt testsystem. Man kan tydligt se en förbättring då man jämför det gamla med det nya testsystemet. Det nya testsystemet har ett bra verktyg för importering av data, vilket gör att förarbetet före testningen effektiveras. All data som importeras och loggas, sparas i datablock. Detta betyder att informationen sparas i PLC:ns minne så länge det finns ett back-up batteri i PLC:n. Tack vare detta kan importerad data och de inställningar som gjorts, sparas i testsystemet trots att testsystemet stängs av.

Enskilda element i testsystemet kan verka simpla, men på grund av att testsystemet är PLC-baserat kan även komplexa sekvenser programmeras. Detta var inte möjligt med det gamla testsystemet. Nu kan även förändringar i testsystemets analoga utgångar ses på ett lätt och överskådligt sätt tack vare de dynamiska staplarna som finns i utgångarnas användargränssnitt. Placeringen av objekten planerades noggrant, men faceplate-instansernas layout ändrades flera gånger innan den slutliga versionen togs i bruk.

Test av säkerhetsautomation till kraftverk är väldigt viktigt, därför bör det finnas bra testutrustning. Enskilda element ur testsystemet kan verka simpla, men eftersom testsystemet är PLC-baserat kan komplexa sekvenser programmeras. Detta var inte möjligt med uppdragsgivarens tidigare testsystem. Testsystemet ses i figur 27 och figur 28



Figur 27 Testsystemet med dess snabbkontakter.



Figur 28 Maskinvaran i testsystemet.

10 Kritisk granskning och diskussion

TIA Portalen var ett nytt verktyg för mig och det tog en tid att bekanta sig med det. Detta är delvis en orsak till den nästan obefintliga planeringen av systemet. Orsaken till den obefintliga planeringen var främst att den kom i användning via kurser om TIA Portalen. Programmeringen påbörjades i ett tidigt skede där de första projekten var testprojekt där TIA Portalens funktionalitet testades.

TIA Portalen är ännu väldigt ostabilt system på grund av att det är ett nytt utvecklingsverktyg. Detta orsakade en del problem som var tidskrävande eftersom man undersökte fel som egentligen inte var fel utan en bugg i programmet. TIA Portalen var ett ostabilt programmeringsverktyg, programmet kraschade ungefär 50 gånger under projektets gång. Vid varje krasch förlorades givetvis allt som hade gjorts sedan senast man sparat projektet. Ostabiliteten i programmet gör mig lite osäker på hur säkert systemet kommer att fungera och ifall detta blir ett verkligt problem.

Vid test av systemet blev det en miss i maskinvaruplaneringen upptäckt. Detta fel kunde ändå korrigeras med relativt enkla medel. Användargränssnittets sid byten fungerar aningen långsamt. Orsaken till detta är okänd, eftersom datorns prestanda inte är orsaken. Orsaken undersöks i skrivande stund, men kunde eventuellt orsakas av stor trafik till PLC:n vid byte av sida eftersom samtliga värden då ska uppdateras.

Trots att testsystemet är enligt min mening användarvänligt så har inga användarvänlighetstester utförts. Om utseendet förbättras kunde man skapa ett lite modernare utseende. Automatiskt genererade rapporter skulle vara till stor nytta vid dokumentation av FAT Detta finns inte idag och var inte heller med i specifikationen av systemet, men kunde vara nästa steg i utvecklingen.

Under hela projektets gång har jag fått mycket hjälp från uppdragsgivaren vid de problem som uppstått. Motivationen att genomföra projektet har varit hög främst på grund av att man vet att testsystemet verkligen behövs och kommer att användas i en FAT redan under våren 2012.

Källförteckning

Bercun, S. (1999) *How to avoid foolish consistency*. <http://www.scottberkun.com/essays/5-how-to-avoid-foolish-consistency/> (Hämtat: 05.03.2012)

Crispin, A. J. (1997). *Programmable logic controllers and their engineering applications*. 2 uppl. Oxford: McGraw-hill publishing company.

International Electrotechnical Commission. (1997) *International Standard IEC 61355. Classification and designation of document for plants, system and equipment*.

International Electrotechnical Commission. (2006) *International Standard IEC 62381. Automation systems in the process industry – Factory acceptance test (FAT), site acceptance test (SAT) and site integration test (SIT)*.

International Electrotechnical Commission. (2003) *International Standard IEC 61131-3. Programmable controllers – Programming languages*

Johnson, J. (2010) *Designing with the mind in mind – Simple guide to understanding user interface design rules*. Burlington: Morgan Kaufmann Publishers.

Joronen, T., Kovács, J. & Majanne, Y. (2007). *Voimalaitosautomaatio* Helsingfors: Suomen Automaatioseura ry

Luukkainen, T., Raivio, T. & Väliäho, T. (2007). *Meikkaus*. Helsinki: WSOYpro Oppimateriaalit.

Moller, M. (2007) *Human Factors HMI Requirements*
http://www.dlr.de/emma/temmaDocs/doc-SP1/D136u_HMI_V1.0.pdf (Hämtat: 01.03.2012)

Siemens. (2011a). *STEP 7 Professional V11.0 SP2*. System manual.
<http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=47071380&subtype=133300&caller=view> (Hämtat: 5.1.2012)

Siemens (2011b). *SIMATIC STEP 7 in the Totally Integrated Automation Portal*. Broschyr.
http://www.siemens.fi/pool/products/industry/iadt_is/tuotteet/tia_portal/simatic-step-7-in-the-tia-portal.pdf (Hämtat: 08.03.2012)

Siemens (2011c). *SIMATIC WinCC in the Totally Integrated Automation Portal*. Broschyr.
http://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure_simatic-wincc_tia-portal_en.pdf (Hämtat: 08.03.2012)